



**TUGAS AKHIR - TE 145561**

**PROTOTYPE SISTEM PEMBAYARAN DENGAN METODE PASCA  
BAYAR MENGGUNAKAN RFID DENGAN MEKANISME PINTU  
GERBANG PADA WAHANA TAMAN BERMAIN**

Muhammad Suprayogi  
NRP 2213030040  
Achmad Nurhisyam  
NRP 2213030045

Dosen Pembimbing 1  
Ir. Djoko Suprajitno Rahardjo, MT.

Dosen Pembimbing 2  
Slamet Budiprayitno, ST. MT.

PROGRAM STUDI D3 TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



***FINAL PROJECT - TE 145561***

***PROTOTYPE OF POSTPAID PAYMENT SYSTEM USING RFID  
WITH GATE MECHANISM FOR THEME PARK***

Muhammad Suprayogi  
NRP 2213030040  
Achmad Nurhisyam  
NRP 2213030045

*Supervisor 1*  
Ir. Djoko Suprajitno Rahardjo, MT.

*Supervisor 2*  
Slamet Budiprayitno, ST. MT.

***ELECTRICAL ENGINEERING D3 STUDY PROGRAM***  
*Faculty of Industrial Technology*  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

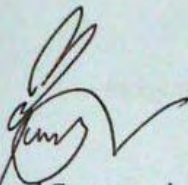


## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Prototipe Sistem Pembayaran dengan Metode Pasca Bayar menggunakan RFID pada Mekanisme Pintu Gerbang Wahana Taman Bermain**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

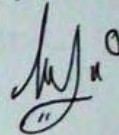
Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Muhammad Suprayogi  
2213030040

Surabaya, 01 Juni 2016



Achmad Nurhisyam  
2213030045

-----Halaman ini sengaja dikosongkan-----

**PROTOTYPE SISTEM PEMBAYARAN DENGAN METODE  
PASCA BAYAR MENGGUNAKAN RFID PADA MEKANISME  
PINTU GERBANG WAHANA TAMAN BERMAIN**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya Teknik  
Pada  
Bidang Studi Komputer Kontrol  
Program Studi D3 Teknik Elektro  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing 1

Dosen Pembimbing 2



Ir. Djoko Suprajitno Rahardjo, MT.  
NIP. 19550622 198701 1 001

Slamet Budiprayitno, ST. MT.  
NIP. 19781113 201012 1 002

**SURABAYA  
JUNI, 2016**

-----Halaman ini sengaja dikosongkan-----

**LEMBAR PERNYATAAN  
PERSETUJUAN PUBLIKASI KARYA ILMIAH  
UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Institut Teknologi Sepuluh Nopember Surabaya, yang bertanda tangan di bawah ini saya :

Nama : Muhammad Suprayogi / Achmad Nurhisyam  
Nrp. : 2213030090 / 2213030095  
Jurusan / Fak. : D3. Teknik Elektro / Fakultas Teknologi Industri  
Alamat kontak : Jl. Marebahan V/R GEB Gresik  
a. Email : moh.suprayogi@gmail.com  
b. Telp/HP : 085621877670

Menyatakan bahwa semua data yang saya *upload* di Digital Library ITS merupakan hasil final (revisi terakhir) dari karya ilmiah saya yang sudah disahkan oleh dosen penguji. Apabila dikemudian hari ditemukan ada ketidaksesuaian dengan kenyataan, maka saya bersedia menerima sanksi.

Demi perkembangan ilmu pengetahuan, saya menyetujui untuk memberikan **Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalti-Free Right)** kepada Institut Teknologi Sepuluh Nopember Surabaya atas karya ilmiah saya yang berjudul :

Prototipe Sistem Pembayaran dengan Metode Pasca Bayar menggunakan RFID dengan Mekanisme Pinis Gerbang pada Wahana Taman Bermain

Dengan Hak Bebas Royalti Non-Eksklusif ini, Institut Teknologi Sepuluh Nopember Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya, dan menampilkan/mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta. Saya bersedia menanggung secara pribadi, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya Ilmiah saya ini tanpa melibatkan pihak Institut Teknologi Sepuluh Nopember Surabaya.

Demikian pernyataan ini saya buat dengan sebenarnya.



NIP. 19550622 198701 1 001

Dibuat di : Surabaya  
Pada tanggal : 28 Juni 2016  
Yang menyatakan :

Muhammad Suprayogi

Nrp. 2213030090

**KETERANGAN :**

Tanda tangan pembimbing wajib dibubuhi stempel jurusan.

Form dicetak dan diserahkan di bagian Pengadaan saat mengumpulkan hard copy TA/Tesis/Disertasi.



# PROTOTYPE SISTEM PEMBAYARAN DENGAN METODE PASCA BAYAR MENGGUNAKAN RFID PADA MEKANISME PINTU GERBANG WAHANA TAMAN BERMAIN

Nama Mahasiswa : Muhammad Suprayogi  
NRP : 2213 030 040  
Nama Mahasiswa : Achmad Nurhisyam  
NRP : 2213 030 045  
Pembimbing 1 : Ir. Djoko Suprajitno Rahardjo, MT.  
NIP : 19550622 198701 1 001  
Pembimbing 2 : Slamet Budiprayitno, ST. MT.  
NIP : 19781113 201012 1 002

## ABSTRAK

Era Globalisasi yang telah meningkatkan pertumbuhan industri turut serta meningkatkan teknologi. Dengan kemajuan teknologi tersebut, salah satunya ditemukan RFID (*Radio Frequency Identification*). RFID adalah bentuk umum teknologi yang menggunakan *radio waves* untuk mengidentifikasi manusia atau objek secara otomatis.

Dalam Tugas Akhir ini dirancang suatu sistem pembayaran pasca bayar pada wahana taman bermain dengan menggunakan sensor RFID yang dapat mengurangi jumlah antrian pengunjung. Sistem ini menggunakan pengontrol berbasis mikrokontroler. *ID* pada *tag* RFID akan dibaca RFID *reader* kemudian akan diproses oleh mikrokontroler. Biaya yang dibutuhkan untuk masuk wahana permainan akan muncul di LCD, kemudian palang pintu wahana permainan akan terbuka. Mikrokontroler di setiap wahana permainan berperangan sebagai *slave* menggunakan komunikasi I<sup>2</sup>C dan terhubung dengan mikrokontroler pada loket pembayaran yang berperan sebagai *master*.

Dari hasil pengujian sistem secara keseluruhan dapat diketahui bahwa sensor mampu mendeteksi adanya *tag* RFID dalam jarak antara 0-4cm. Selain itu untuk nilai transaksi pada wahana permainan tidak melebihi nominal Sembilan ratus sembilan puluh sembilan ribu rupiah.

**Kata Kunci :** RFID, Wahana Taman Bermain, Mikrokontroler, Motor *Servo*, LCD.

-----Halaman ini sengaja dikosongkan-----

**PROTOTYPE OF POSTPAID PAYMENT SYSTEM USING RFID  
WITH GATE MECHANISM FOR THEME PARK**

**Name** : Muhammad Suprayogi  
**Registration Number** : 2213030040  
**Name** : Achmad Nurhisyam  
**Registration Number** : 2213030045  
**Supervisor 1** : Ir. Djoko Suprajitno Rahardjo, MT.  
**ID** : 19550622 198701 1 001  
**Supervisor 2** : Slamet Budiprayitno, ST. MT.  
**ID** : 19781113 201012 1 002

**ABSTRACT**

*In era of globalization, technological development is increasing, especially in the industrial field. It was marked by discovery of RFID (Radio Frequency Identification). RFID can identify people or object automatically.*

*In this final project designed a postpaid payment system using RFID for theme park. This system can reduce a number of queues on the visitors. The system uses microcontroller. ID on the RFID tag is read by RFID reader and processed by microcontroller. Cost to enter the playground will appear on LCD. Then portal will open playground door. Microcontroller on the playground acts as a slave, while the microcontroller at payment counter acts as a master.*

*From the test result known that sensor can detect RFID tag at a distance of 0-4cm. The value of transactions on the playground is not more than a nominal nine hundred ninety nine thousand rupiah.*

**Keywords** : RFID, Theme Park, Microcontroller, Servo Motor, LCD

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

## HALAMAN

HALAMAN JUDUL .....	i
HALAMAN JUDUL .....	ii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	ii
HALAMAN PENGESAHAN .....	iv
ABSTRAK .....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR .....	<b>Error! Bookmark not defined.</b>
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL .....	xv
<b>BAB I PENDAHULUAN .....</b>	<b>Error! Bookmark not defined.</b>
1.1 Latar Belakang.....	<b>Error! Bookmark not defined.</b>
1.2 Permasalahan .....	<b>Error! Bookmark not defined.</b>
1.3 Batasan Masalah.....	<b>Error! Bookmark not defined.</b>
1.4 Tujuan.....	<b>Error! Bookmark not defined.</b>
1.5 Metodologi Penelitian.....	<b>Error! Bookmark not defined.</b>
1.6 Sistematika Laporan .....	<b>Error! Bookmark not defined.</b>
1.7 Relevansi .....	<b>Error! Bookmark not defined.</b>
<b>BAB II TEORI DASAR.....</b>	<b>Error! Bookmark not defined.</b>
2.1 Sistem Pasca Bayar .....	<b>Error! Bookmark not defined.</b>
2.2 RFID ( <i>Radio Frequency Identification</i> ).....	<b>Error! Bookmark not defined.</b>
2.2.1 Sistem RFID .....	<b>Error! Bookmark not defined.</b>
2.2.2 Prinsip Kerja RFID .....	<b>Error! Bookmark not defined.</b>
2.2.3 Beberapa Tipe <i>tag</i> RFID ..	<b>Error! Bookmark not defined.</b>
2.2.4 Frekuensi Radio dan Jangkauan.....	<b>Error! Bookmark not defined.</b>
2.3 Teori Dasar LCD.....	<b>Error! Bookmark not defined.</b>
2.4 Motor <i>Servo</i> .....	<b>Error! Bookmark not defined.</b>
2.4.1 Prinsip Kerja Motor <i>Servo</i> ..	<b>Error! Bookmark not defined.</b>
2.4.2 Fungsi Potensiometer pada Motor <i>Servo</i> .....	<b>Error! Bookmark not defined.</b>
2.5 Mikrokontroler ATmega328.....	<b>Error! Bookmark not defined.</b>
2.6 <i>Power Supply</i> .....	<b>Error! Bookmark not defined.</b>
2.7 Komunikasi I2C ( <i>Inter Integrated Circuit</i> ).....	<b>Error! Bookmark not defined.</b>

2.8	Hukum Newton .....	<b>Error! Bookmark not defined.</b>
2.9	Pesawat Sederhana (Tuas).....	<b>Error! Bookmark not defined.</b>
<b>BAB III PERANCANGAN SISTEM KONTROL</b>		<b>Error! Bookmark not defined.</b>
<b>not defined.</b>		
3.1	Perancangan Elektronika.....	<b>Error! Bookmark not defined.</b>
3.1.1	Cara Kerja Sistem.....	<b>Error! Bookmark not defined.</b>
3.1.2	Modul <i>Regulator</i> LM2596	<b>Error! Bookmark not defined.</b>
3.1.3	Rangkaian Mikrokontroler ATmega328	<b>Error! Bookmark not defined.</b>
3.1.4	<i>Liquid Cristal Display</i> (LCD)	<b>Error! Bookmark not defined.</b>
3.1.5	Rangkaian RFID.....	<b>Error! Bookmark not defined.</b>
3.1.6	Rangkaian Motor <i>Servo</i> ....	<b>Error! Bookmark not defined.</b>
3.1.7	Rangkaian Komunikasi I2C	<b>Error! Bookmark not defined.</b>
3.1.8	<i>Power Supply</i> .....	<b>Error! Bookmark not defined.</b>
3.2	Perancangan Perangkat Lunak....	<b>Error! Bookmark not defined.</b>
3.2.1	Aplikasi Arduino <i>Integrated Development Environment</i> (IDE).....	<b>Error! Bookmark not defined.</b>
3.2.2	Pembuatan <i>Flowchart</i> Program	<b>Error! Bookmark not defined.</b>
3.2.3	Perancangan Perangkat Lunak untuk RFID	<b>Error! Bookmark not defined.</b>
3.2.4	Perancangan <i>Software</i> Pengelola <i>Database</i>	<b>Error! Bookmark not defined.</b>
3.3	Perancangan Mekanik.....	<b>Error! Bookmark not defined.</b>
3.3.1	Pintu Wahana.....	<b>Error! Bookmark not defined.</b>
3.3.2	Palang Pintu.....	<b>Error! Bookmark not defined.</b>
3.3.3	<i>Packaging</i> Rangkaian .....	<b>Error! Bookmark not defined.</b>
<b>BAB IV HASIL SIMULASI DAN IMPLEMENTASI</b>		<b>Error! Bookmark not defined.</b>
<b>not defined.</b>		
4.1	Pengujian pada RFID RC522 .....	<b>Error! Bookmark not defined.</b>
4.2	Pengujian LCD.....	<b>Error! Bookmark not defined.</b>
4.3	Pengujian Keseluruhan .....	<b>Error! Bookmark not defined.</b>
<b>BAB V PENUTUP</b>		1
5.1	Kesimpulan .....	1
5.2	Saran .....	1
<b>DAFTAR PUSTAKA</b>		3
<b>LAMPIRAN A</b>		<b>Error! Bookmark not defined.</b>
<b>LAMPIRAN B</b>		<b>Error! Bookmark not defined.</b>
<b>LAMPIRAN C</b>		<b>Error! Bookmark not defined.</b>

DAFTAR RIWAYAT HIDUP .....	5
----------------------------	---

## DAFTAR GAMBAR

### HALAMAN

<b>Gambar 2.1</b>	<b>RFID Reader RC522.....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 2.2</b>	<b>Motor Servo.....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 2.3</b>	<b>Prinsip Kerja Motor Servo.</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 2.4</b>	<b>ATMega328 .....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 2.5</b>	<b>Kondisi Sinyal Start dan Stop</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 2.6</b>	<b>Sinyal ACK dan NACK....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 2.7</b>	<b>Transfer Bit pada I2C Bus.</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.1</b>	<b>Alur Sistematika pada Wahana Permainan</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.2</b>	<b>Diagram Fungsional Alat ..</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.3</b>	<b>Modul Regulator LM2596</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.4</b>	<b>Fungsional Mikrokontroler ATMega328</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.5</b>	<b>Port I/O ATMega328.....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.6</b>	<b>Fungsional LCD .....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.7</b>	<b>Fungsional RFID .....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.8</b>	<b>RFID Tag .....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.9</b>	<b>RFID RC522 Pin Out.....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.10</b>	<b>Skematik RFID Reader RC522</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.11</b>	<b>Rangkaian RFID .....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.12</b>	<b>Fungsional Motor Servo ..</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.13</b>	<b>Rangkaian Motor Servo dengan Proteus</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.14</b>	<b>Rangkaian Komunikasi I2C dengan Proteus</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.15</b>	<b>Setting Serial Port.....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.16</b>	<b>Flowchart Master Sistem Wahana</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.17</b>	<b>Flowchart Slave Sistem Wahana</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.18</b>	<b>Desain Pintu Wahana .....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 3.19</b>	<b>Desain Packaging Rangkaian</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 4.1</b>	<b>Program untuk Cek Data Kartu</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 4.2</b>	<b>ID pada RFID Tag .....</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 4.3</b>	<b>Pengukuran RFID menggunakan Multimeter</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 4.4</b>	<b>Perbandingan Jarak RFID Tag 5A 7E F6 A5</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 4.5</b>	<b>Perbandingan Jarak RFID Tag B0 64 70 B5</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 4.6</b>	<b>Perbandingan Jarak RFID Tag B0 64 70 B5</b>	<b>Error! Bookmark not defined.</b>
<b>Gambar 4.7</b>	<b>Perbandingan Jarak RFID Tag 84 B3 1F 3B</b>	<b>Error! Bookmark not defined.</b>



**Gambar 4.8** Perbandingan Jarak RFID *Tag* E0 50 C2 1E **Error! Bookmark not defined.**

**Gambar 4.9** Program Perintah pada LCD **Error! Bookmark not defined.**

**Gambar 4.10** Rangkaian LCD dengan *Proteus* **Error! Bookmark not defined.**

**Gambar 4.11** Program untuk Mendeteksi *Tag* dan Ditampilkan di  
LCD ..... **Error! Bookmark not defined.**

**Gambar 4.12** Program Motor *Servo*..... **Error! Bookmark not defined.**

## DAFTAR TABEL

### HALAMAN

<b>Tabel 2.1</b> Fungsi <i>Pin</i> LCD .....	<b>Error! Bookmark not defined.</b>
<b>Tabel 2.2</b> Konfigurasi <i>Port B</i> .....	<b>Error! Bookmark not defined.</b>
<b>Tabel 2.3</b> Konfigurasi <i>Port C</i> .....	<b>Error! Bookmark not defined.</b>
<b>Tabel 2.4</b> Konfigurasi <i>Port D</i> .....	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.1</b> RFID <i>Reader</i> pada Wahana 1	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.2</b> RFID <i>Reader</i> pada Wahana 2	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.3</b> RFID <i>Reader</i> pada Loket Pembayaran	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.4</b> RFID <i>Reader</i> pada Wahana 1	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.5</b> RFID <i>Reader</i> pada Wahana 2	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.6</b> RFID <i>Reader</i> pada Loket Pembayaran	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.7</b> RFID <i>Reader</i> pada Wahana 1	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.8</b> RFID <i>Reader</i> pada Wahana 2	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.9</b> RFID <i>Reader</i> pada Loket Pembayaran	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.10</b> RFID <i>Reader</i> pada Wahana 1	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.11</b> RFID <i>Reader</i> pada Wahana 2	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.12</b> RFID <i>Reader</i> pada Loket Pembayaran	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.13</b> RFID <i>Reader</i> pada Wahana 1	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.14</b> RFID <i>Reader</i> pada Wahana 2	<b>Error! Bookmark not defined.</b>
<b>Tabel 4.15</b> RFID <i>Reader</i> pada Loket Pembayaran	<b>Error! Bookmark not defined.</b>

-----Halaman ini sengaja dikosongkan-----



-----Halaman ini sengaja dikosongkan-----

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam kehidupan masyarakat modern, manusia dapat dengan mudah merasakan gejala *stress* yang diakibatkan oleh tuntutan hidup dan padatnya jadwal pekerjaan. Hal tersebut mengakibatkan kebutuhan akan rekreasi dan hiburan semakin meningkat. Taman Bermain adalah salah satu tempat yang banyak didatangi oleh masyarakat saat sedang membutuhkan hiburan. Wahana yang disediakan beragam seperti *roller coaster*, *thunder*, dan *tornado*. Perubahan perilaku dan cara pandang masyarakat untuk mengunjungi taman bermain sudah menjadi bagian dari *life style*.

Pada beberapa wahana taman bermain, pengunjung perlu melakukan pembayaran pada loket untuk menikmati wahana permainan yang ada. Selain itu di beberapa wahana yang disediakan ada yang sudah menggunakan sistem RFID namun dengan metode Pra Bayar, sehingga hal tersebut menyebabkan antrian di tempat pengisian saldo pada loket registrasi di pintu masuk taman bermain. Hal ini menyebabkan pengunjung kelelahan sebelum menikmati wahana permainan. Dampak lain adalah menyebabkan banyaknya calon pengunjung yang membatalkan niat untuk memasuki wahana taman bermain. Oleh karena itu pada Tugas Akhir ini dibuat suatu rancangan sistem pembayaran dengan metode pasca bayar yang diterapkan pada wahana taman bermain dengan menggunakan sensor RFID.

Teknologi RFID yang digunakan terdiri dari dua komponen utama yaitu *tag* RFID dan *RFID reader*. *ID* pada *tag* RFID akan dibaca *RFID reader* kemudian akan diproses oleh mikrokontroler. Biaya yang dibutuhkan untuk masuk wahana permainan akan muncul di LCD, kemudian palang pintu wahana permainan akan terbuka. Sistem pembayaran ini menggunakan komunikasi I<sup>2</sup>C, dimana mikrokontroler di setiap wahana permainan berperan sebagai *slave* dan terhubung dengan mikrokontroler pada loket pembayaran yang berperan sebagai *master*. Pada sistem pembayaran menggunakan RFID, pengunjung dapat lebih mudah melakukan transaksi pembayaran sehingga antrian yang sering terjadi dapat di minimalisir.

## **1.2 Permasalahan**

Pada Tugas Akhir ini yang menjadi permasalahan utama adalah proses transaksi pembelian tiket yang menyebabkan antrian panjang pada wahana taman bermain.

## **1.3 Batasan Masalah**

Dari permasalahan di atas, maka batasan masalah pada Tugas Akhir ini yaitu terletak pada desain taman bermain sebagai alat untuk pengambilan data kami rancang menjadi 2 tempat wahana saja. Selain itu nominal transaksi pada masing-masing wahana tidak melebihi nominal Sembilan ratus isbandi puluh isbandi ribu rupiah.

## **1.4 Tujuan**

Tugas Akhir ini bertujuan untuk membuat *prototype* sistem pembayaran dengan metode pasca bayar menggunakan RFID dengan mekanisme pintu gerbang pada taman bermain sehingga antrian yang sering terjadi pada saat proses transaksi pembayaran dapat diminimalisir.

## **1.5 Metodologi Penelitian**

Penelitian ini dilakukan melalui beberapa tahapan metodologi yaitu, studi isbanding, pemodelan sistem, simulasi hasil desain, implementasi dan analisis data, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi isbanding akan dipelajari mengenai identifikasi fisik, dan cara kerja dari sensor RFID. Pada tahap pemodelan sistem akan dibentuk dari hasil identifikasi fisik. Yang terakhir adalah merancang hasil pemodelan sistem dan simulasi hasil desain, implementasi, dan melakukan pengambilan data. Data yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

## **1.6 Sistematika Laporan**

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

**Bab I      Pendahuluan**

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

**Bab II      Teori Dasar**

Bab ini menjelaskan tentang sistem pasca bayar, RFID (*Radio Frequency Identification*), teori dasar LCD, Motor Servo, Mikrokontroler ATmega328, Power Supply, Komunikasi I2C, Hukum Newton, dan Pesawat Sederhana.

**Bab III     Perancangan Sistem**

Bab ini membahas tentang perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*) berdasarkan teori dasar pada Bab II.

**Bab IV     Simulasi, Implementasi dan Analisis Sistem**

Bab ini memuat hasil simulasi dan implementasi serta analisis dari hasil tersebut.

**Bab V      Penutup**

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

**1.7 Relevansi**

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat memberikan manfaat, yaitu mempermudah proses sistem pembayaran pada wahana permainan sehingga lebih efisien dan nyaman. Sistem pembayaran dengan RFID tidak mengharuskan pengunjung untuk melakukan transaksi pembelian tiket, sehingga tidak mengganggu kenyamanan saat bermain.



-----Halaman ini sengaja dikosongkan-----

## **BAB II**

### **TEORI DASAR**

#### **2.1 Sistem Pasca Bayar**

Sistem pasca bayar adalah sistem pembayaran yang dilakukan setelah menggunakan layanan atau servis dari pihak penyedia layanan. Masyarakat umumnya mengenal istilah pasca bayar melalui paket kartu operator telepon seluler. Sistem pasca bayar merupakan salah satu isbanding pengguna selain sistem pra bayar. Nilai plus dari layanan pascabayar adalah tidak adanya batasan waktu dan penggunaan selama si pengguna menaati peraturan yang berlaku. Dalam pemakaian paket ini, penyedia layanan akan memberikan total biaya pada akhir penggunaan. Akan tetapi jika pengguna layanan mengalami tunggakan atau tidak melakukan pembayaran sama sekali, maka penyedia layanan akan memberikan denda sesuai persyaratan sebelumnya.

Pelanggan layanan pascabayar umumnya menggunakan servis jenis ini karena memiliki beberapa keuntungan yang sekaligus menjadi pembeda dengan paket Prabayar. Selain biaya yang ditawarkan lebih murah dan hemat dibandingkan dengan Prabayar, paket pasca bayar juga menyediakan jaminan apabila terjadi gangguan sementara atau permanen pada layanan yang anda gunakan.

Penggunaan paket pascabayar umumnya dihitung berdasarkan waktu penggunaan aktif dan periode di saat paket tersebut sama sekali tidak digunakan. Tagihan dalam paket pascabayar sendiri adalah salah satu komponen utama pada layanan tersebut dimana dapat mewakili eksistensi dari servis operator itu sendiri sebagai bukti pemakaian layanan oleh pengguna paket tersebut. Tagihan pemakaian umumnya harus mudah dibaca, mudah dimengerti dan juga menarik perhatian bagi para pelanggan untuk melihat fitur lain di dalam tagihan tersebut.

#### **2.2 RFID (*Radio Frequency Identification*)**

RFID atau *Radio Frequency Identification*, adalah suatu metode yang bisa digunakan untuk menyimpan atau menerima data secara jarak jauh dengan menggunakan suatu piranti yang bernama *tag* RFID atau *transponder*. Suatu *tag* RFID adalah sebuah benda kecil, dan dapat ditempelkan pada suatu barang atau produk. *Tag* RFID berisi isbanding yang memungkinkan mereka untuk menerima dan merespon terhadap suatu data yang dipancarkan oleh suatu RFID *transceiver*. [1]

### 2.2.1 Sistem RFID[1]

Suatu sistem RFID secara utuh terdiri dari 3 komponen yaitu:

1. *Tag* RFID, dapat berupa stiker, kertas atau isband dengan beragam ukuran. Didalam setiap *tag* terdapat *chip* yang mampu menyimpan sejumlah informasi tertentu.
2. *Reader* RFID, terdiri atas *RFID reader* dan isband yang akan mempengaruhi jarak optimal identifikasi. *Reader* RFID akan membaca atau mengubah informasi yang tersimpan didalam *tag* melalui frekuensi radio. *Reader* RFID terhubung langsung dengan sistem mikrokontroler.
3. Mikrokontroler, bisa berupa arduino. Fungsinya mengatur alur informasi dari item-item yang terdeteksi dalam lingkup sistem RFID dan mengatur komunikasi antara *tag* dan *reader*.



**Gambar 2.1** RFID *Reader* RC522[2]

### 2.2.2 Prinsip Kerja RFID[1]

Teknologi RFID didasarkan pada prinsip kerja gelombang elektromagnetik, dimana:

1. Komponen utama dari *tag* RFID adalah *chip* dan *tag-antena* yang biasa disebut dengan *inlay*, dimana *chip* berisi informasi dan terhubung dengan *tag-antena*.
2. Informasi yang berada/tersimpan dalam *chip* ini akan terkirim/terbaca melalui gelombang elektromagnetik setelah *tag-antena* mendapatkan/menerima pancaran gelombang elektromagnetik dari *reader-antena* (*Interogator*). *RFID reader* ini yang sekaligus akan meneruskan informasi pada *application server*.

### 2.2.3 Beberapa Tipe tag RFID[3]

RFID *tag* dapat bersifat aktif dan pasif. RFID *tag* yang bersifat pasif tidak memiliki *power supply* sendiri. Dengan hanya berbekal induksi listrik yang ada pada isband yang disebabkan oleh adanya frekuensi radio *scanning* yang masuk, sudah cukup isban kekuatan yang cukup bagi RFID *tag* untuk mengirim respon balik. Sehubungan dengan *power* dan biaya, maka respon dari suatu RFID yang bersifat pasif biasanya sederhana, hanya nomer ID saja. Dengan tidak adanya *power supply* pada RFID *tag* yang pasif maka akan menyebabkan semakin kecilnya ukuran dari RFID *tag* yang mungkin dibuat. Beberapa RFID komersial saat ini sudah beredar di pasaran ada yang bisa diletakkan dibawah kulit. RFID *tag* yang pasif memiliki jarak jangkauan yang berbeda mulai dari 10mm sampai dengan 6 meter. RFID *tag* yang bersifat aktif, disisi lalin harus memiliki *power supply* sendiri dan memiliki jarak jangkauan yang lebih jauh. Memori yang dimilikinya juga lebih besar sehingga bisa menampung berbagai macam informasi didalamnya. Ukuran terkecil dari RFID *tag* yang bersifat aktif yaitu seukuran koin. Jarak jangkauan dari RFID *tag* aktif ini bisa sampai sekitar 10 meter dengan umur baterai yang bisa mencapai beberapa tahun lamanya. RFID *tag* pasif memiliki harga yang lebih murah dari RFID *tag* yang bersifat aktif.

### 2.2.4 Frekuensi Radio dan Jangkauan[4]

Ada empat macam RFID *tag* yang sering digunakan berdasarkan frekuensi radio, yaitu:

1. *Low Frequency* RFID *tag* (125-134 Khz)  
Digunakan untuk identifikasi binatang, *beer keg tracking*, *keylock* pada mobil, dan sistem anti pencuri.
2. *High Frequency* RFID *tag* (13, 56 Mhz)  
Digunakan pada perpustakaan atau isb buku, *pallet tracking*, akses kontrol gedung, dan *apparel item tracking*.
3. UHF RFID *tag* (868 sampai 956 Mhz)  
Sering digunakan untuk *pallet* pelacakan isbandin, pelacakan truk, dan *trailer* pada pelabuhan kapal laut.
4. *Microwave* RFID *tag* (2,45 GHz)  
Sering digunakan dalam akses kontrol jarak jauh kendaraan bermotor.

Jarak antara isband pembaca RFID dengan *tag* secara langsung dipengaruhi oleh frekuensi yang digunakan.

Ukuran isband yang digunakan untuk transmisi data bergantung dari panjang gelombang elektromagnetik. Untuk frekuensi yang rendah, maka isband harus dibuat dengan ukuran yang lebih besar dibandingkan dengan RFID frekuensi tinggi.

### 2.3 Teori Dasar LCD[1]

LCD (*Liquid Cristal Display*) merupakan suatu bentuk kristal cair yang akan ber-emulsi apabila diberi tegangan. Tampilannya berupa *dot matrix* 5x7 dengan jenis huruf yang ditampilkan akan lebih banyak dan lebih baik resolusinya jika dibandingkan dengan *seven segment*. LCD yang dipergunakan mempunyai spesifikasi sebagai berikut:

1. Terdiri dari 16 karakter yang tersusun dalam dua baris (masing-masing 8 karakter) dengan *display dot matrix* 5x7.
2. Karakter generator ROM dengan 192 tipe karakter.
3. Karakter generator RAM dengan 8 tipe karakter.
4. 80x8 *bit display* data RAM.
5. Catu daya +5 Volt.
6. *Reset* pada saat *Power On*.

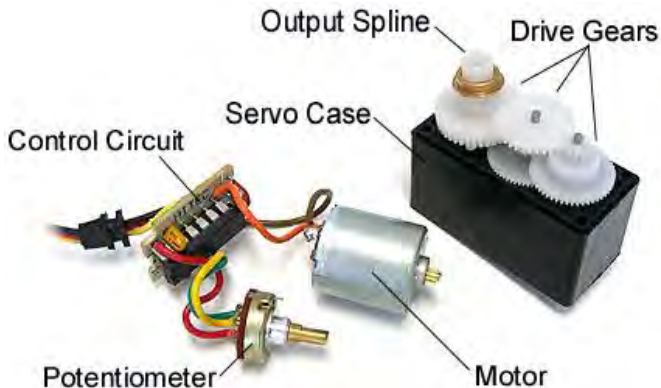
**Tabel 2.1** Fungsi *Pin* LCD

No. <i>Pin</i>	Nama <i>Pin</i>	Fungsi
<i>Pin</i> 16	V – BL	Sebagai <i>ground</i> dari <i>backlight</i>
<i>Pin</i> 15	V + BL	Sebagai kutub positif dari <i>backlight</i>
<i>Pin</i> 7 – <i>Pin</i> 14	DB0 – DB7	Merupakan saluran data, berisi perintah dan data akan ditampilkan
<i>Pin</i> 6	E	Sinyal operasi awal, sinyal ini mengaktifkan data tulis atau baca
<i>Pin</i> 5	R/W	Sinyal seleksi tulis atau baca 0 = tulis 1 = baca
<i>Pin</i> 4	RS	Sinyal pemilih <i>register</i> 0 = instruksi <i>register</i> (tulis) 1 = data <i>register</i> (tulis dan baca)
<i>Pin</i> 3	Vlc	Untuk mengendalikan kecerahan LCD dengan merubah Vlc
<i>Pin</i> 2	Vcc	Tegangan catu +5 volt
<i>Pin</i> 1	Vss	Terminal <i>Ground</i>

## 2.4 Motor Servo[5]

Motor *servo* adalah sebuah motor DC dengan sistem umpan balik tertutup dimana posisi rotornya akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor *servo*. Motor ini terdiri dari sebuah motor DC, serangkaian gear yang berfungsi memperlambat putaran dan meningkatkan torsi, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran *servo*. Sedangkan sudut dari sumbu motor *servo* diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor *servo*.

Penggunaan sistem kontrol *loop* tertutup pada motor *servo* berguna untuk mengontrol gerakan dan posisi akhir dari poros motor *servo*. Motor *servo* biasa digunakan dalam aplikasi-aplikasi isbandi, selain itu juga digunakan dalam berbagai aplikasi lain seperti pada mobil mainan radio kontrol, *robot*, dan pesawat.



**Gambar 2.2** Motor Servo[6]

Ada dua jenis motor *servo*, yaitu motor *servo* AC dan DC. Motor *servo* AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin isbandi. Sedangkan motor *servo* DC biasanya lebih cocok untuk digunakan pada aplikasi-aplikasi yang lebih kecil. Dan bila dibedakan menurut rotasinya, terdapat dua jenis motor *servo* yang terdapat di pasaran, yaitu motor *servo rotation 180°* dan *servo rotation continous*.

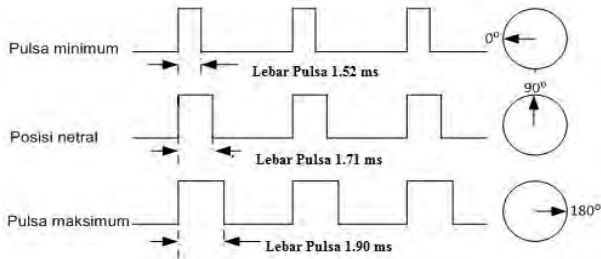
Motor *servo standard (servo rotation 180°)* adalah jenis yang paling umum dari motor *servo*, dimana putaran poros *output* terbatas

hanya  $90^\circ$  kearah kanan dan  $90^\circ$  kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau  $180^\circ$

Motor *servo rotation continous* merupakan jenis motor *servo* yang sebenarnya sama dengan jenis motor *servo standard*, hanya saja perputaran porosnya tanpa batasan atau dengan kata lain dapat berputar terus, baik kearah kanan maupun kiri.

### 2.4.1 Prinsip Kerja Motor Servo

Motor *servo* dikendalikan dengan memberikan sinyal modulasi lebar pulsa (*Pulse Wide Modulation / PWM*) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor *servo*. Sebagai contoh, lebar pulsa dengan waktu 1,71 ms maka akan memutar poros motor *servo* ke posisi  $90^\circ$ . Bila pulsa lebih pendek dari 1,71 ms maka akan berputar isband posisi  $0^\circ$  atau ke kiri (berlawanan jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1,71 ms maka poros motor *servo* akan berputar kea rah posisi  $180^\circ$  atau ke kanan (searah jarum jam).



**Gambar 2.3** Prinsip Kerja Motor Servo[7]

Ketika lebar pulsa kendali telah diberikan, maka poros motor *servo* akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka motor *servo* akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (*rating torsi servo*). Namun motor *servo* tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms untuk menginstruksikan agar posisi poros motor *servo* tetap bertahan pada posisinya.

### 2.4.2 Fungsi Potensiometer pada Motor Servo

Potensiometer pada motor *servo* berfungsi merubah resistansi motor *servo* ketika sudah pada sudut yang diinginkan, sehingga arus berubah dan berpengaruh pada lebar sinyal pulsa yang diinginkan.

### 2.5 Mikrokontroler ATmega328[8]

ATmega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*). Beberapa tipe mikrokontroler yang sama dengan ATmega328 ini antara lain ATmega8535, ATmega16, dan ATmega32, yang membedakannya adalah ukuran memori, banyaknya GPIO (*General Purpose Input/Output*), *peripheral* (USART, *timer*, *counter*, dll). Dari segi ukuran fisik, ATmega328 memiliki ukuran fisik lebih kecil dibandingkan dengan beberapa mikrokontroler lainnya. Mikrokontroler ini memiliki beberapa fitur antara lain:

1. 130 macam instruksi yang isban semuanya dieksekusi dalam satu siklus *clock*
2. 32 x 8-bit *register* serba guna
3. Kecepatan mencapai 16 MIPS dengan *clock* 16 MHz
4. 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari *flash* memori sebagai *bootloader*
5. Memiliki *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 1 KB sebagai tempat penyimpanan data semi permanen karena *EEPROM* tetap dapat menyimpan data meskipun catu daya dimatikan.
6. Memiliki *SRAM* (*Static Random Acces Memory*) sebesar 2KB
7. Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya *PWM* (*Pulse Width Modulation*)
8. *Master/Slave SPI Serial interface*



**Gambar 2.4** ATmega328[9]



ATMega328 memiliki 3 buah *PORT* utama yaitu *PORT B*, *PORT C*, dan *PORT D* dengan total pin *input/output* sebanyak 23 pin. *PORT* tersebut dapat difungsikan sebagai *input/output* digital atau difungsikan sebagai peripheral lainnya.

**Tabel 2.2** Konfigurasi *Port B*

<b><i>Port Pin</i></b>	<b><i>Alternate Function</i></b>
PB7	XTAL2 ( <i>Chip Clock Oscillator pin 2</i> ) TOSC2 ( <i>Timer Oscillator pin 2</i> ) PCINT7 ( <i>Pin Change Interrupt 7</i> )
PB6	XTAL1 ( <i>Chip Clock Oscillator pin 1 or External clock input</i> ) TOSC1 ( <i>Timer Oscillator pin 1</i> ) PCINT6 ( <i>Pin Change Interrupt 6</i> )
PB5	SCK ( <i>SPI Bus Master Input/Slave Output</i> ) PCINT5 ( <i>Pin Change Interrupt 5</i> )
PB4	MISO ( <i>SPI Bus Master Output/Slave Output</i> ) PCINT4 ( <i>Pin Change Interrupt 4</i> )
PB3	MOSI ( <i>SPI Bus Master Output/Slave Input</i> ) OC2A ( <i>Timer/Counter2 Output Compare Match A Output</i> ) PCINT3 ( <i>Pin Change Interrupt 3</i> )
PB2	SS ( <i>SPI Bus Master Slave select</i> ) OC1B ( <i>Timer/Counter1 Output Compare Match 1 Output</i> ) PCINT2 ( <i>Pin Change Interrupt 2</i> )
PB1	OC1A ( <i>Timer/Counter1 Output Compare Match A Output</i> ) PCINT1 ( <i>Pin Change Interrupt 1</i> )
PB0	ICP1 ( <i>Timer/Counter1 Input Capture Input</i> ) CLKO ( <i>Divided System Clock Output</i> ) PCINT0 ( <i>Pin Change Interrupt 0</i> )

**Tabel 2.3** Konfigurasi *Port C*

<b><i>Port Pin</i></b>	<b><i>Alternate Function</i></b>
PC6	RESET ( <i>Reset pin</i> ) PCINT14 ( <i>Pin Change Interrupt 14</i> )
PC5	ADC5 ( <i>ADC Input Channel 5</i> ) SCL ( <i>2-wire Serial Bus Clock Line</i> ) PCINT13 ( <i>Pin Change Interrupt 13</i> )

<b>Port Pin</b>	<b>Alternate Function</b>
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 ( ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

**Tabel 2.4** Konfigurasi Port D

<b>Port Pin</b>	<b>Alternate Function</b>
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

## 2.6 *Power Supply*[10]

*Power Supply* adalah alat atau perangkat keras yang mampu menyuplai tenaga atau tegangan listrik secara langsung dari sumber tegangan listrik yang lainnya. *Power Supply* atau dalam Bahasa Indonesia disebut dengan Catu Daya dapat menyediakan isban listrik yang kemudian mengubahnya menjadi isban listrik yang dibutuhkan oleh perangkat elektronika lainnya. Oleh karena itu, *Power Supply* kadang-kadang disebut juga dengan istilah *Electrical Power Converter*.

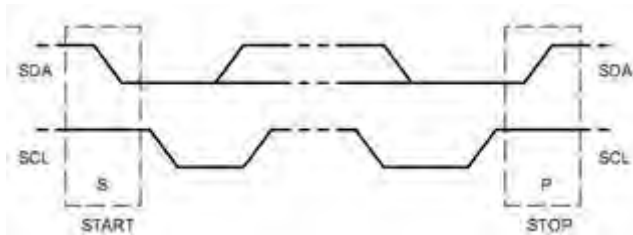
Berdasarkan fungsinya, *Power Supply* dapat dibedakan menjadi *Regulated Power Supply*, *Unregulated Power Supply*, dan *Adjustable Power Supply*.

1. *Regulated Power Supply* adalah *Power Supply* yang dapat menjaga kestabilan tegangan dan arus listrik meskipun terdapat perubahan atau variasi pada beban atau sumber listrik (Tegangan dan Arus *input*).
2. *Unregulated Power Supply* adalah *Power Supply* tegangan ataupun arus listriknya dapat berubah ketika beban berubah atau sumber listriknya mengalami perubahan.
3. *Adjustable Power Supply* adalah *Power Supply* yang tegangan atau arusnya dapat diatur sesuai kebutuhan dengan menggunakan Knob Mekanik.

## 2.7 *Komunikasi I2C (Inter Integrated Circuit)*[11]

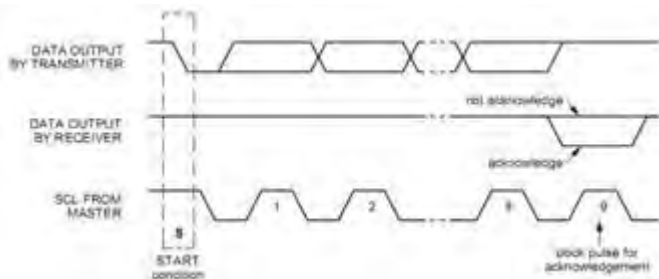
*Inter Integrated Circuit* atau sering disebut I2C adalah standart komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa sistem informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *master* dan *slave*. *Master* adalah piranti yang melalui *transfer* data pada I2C Bus dengan membentuk sinyal *start*. Mengakhiri *transfer* data dengan membentuk sinyal *stop*, dan membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati *master*.

Sinyal *start* adalah sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari "1" menjadi "0" pada SCL "1". Sinyal *stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari "0" menjadi "1" pada SCL "1". Kondisi sinyal *start* dan *stop* seperti tampak pada Gambar 2.5



**Gambar 2.5** Kondisi Sinyal *Start* dan *Stop*[12]

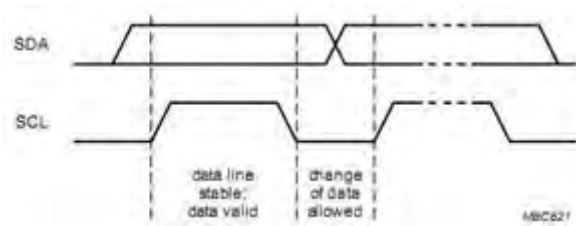
Sinyal dasar yang lain dalam I2C Bus adalah sinyal *acknowledge* yang disimbolkan dengan ACK. Setelah transfer data oleh *master* berhasil diterima *slave*, *slave* akan menjawabnya dengan mengirim sinyal *acknowledge*, yaitu dengan membuat SDA menjadi “0” selama siklus *clock* ke 9. Ini menunjukkan bahwa *slave* telah menerima 8 bit data dari *master*. Kondisi sinyal *acknowledge* seperti Gambar 2.6



**Gambar 2.6** Sinyal ACK dan NACK[13]

Dalam melakukan *transfer* data pada I2C Bus, kita harus mengikuti tata cara yang telah ditetapkan yaitu:

1. *Transfer* data hanya dapat dilakukan ketika Bus tidak dalam keadaan sibuk.
2. Selama proses *transfer* data, keadaan data pada SDA harus stabil selama SCL dalam keadaan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah.



**Gambar 2.7** Transfer *Bit* pada I2C Bus[14]

## 2.8 Hukum Newton[8]

Hukum Newton menggambarkan hubungan antara gaya yang bekerja di suatu benda dan gerak yang disebabkan. Ada 3 Hukum Newton yaitu:

### 1. Hukum Newton I

Bunyi: *“Jika resultan gaya yang bekerja pada benda yang sama dengan nol, maka benda yang mula-mula diam akan tetap diam. Benda yang mula-mula bergerak lurus beraturan akan tetap lurus beraturan dengan kecepatan tetap”*.

Rumus Hukum Newton I yaitu:

$$\Sigma F = 0 \quad (2.1)$$

### 2. Hukum Newton II

Bunyi: *“Percepatan dari suatu benda akan sebanding dengan jumlah gaya (resultan gaya) yang bekerja pada benda tersebut dan berbanding terbalik dengan massanya”*.

Rumus Hukum Newton II yaitu:

$$a = \frac{\Sigma F}{m} \text{ atau } \Sigma F = m \times a \quad (2.2)$$

Keterangan:

$\Sigma F$  = Resultan Gaya (N)

$F$  = Gaya (N)

$m$  = Masa (kg)

$a$  = Percepatan ( $m/s^2$ )

### 3. Hukum Newton III

Bunyi: “Jika suatu benda memberikan gaya pada benda lain maka benda yang dikenai gaya akan memberikan gaya yang besarnya sama dengan gaya yang diterima dari benda pertama tetapi arahnya berlawanan”.

Rumus Hukum III yaitu:

#### 1. Gaya Gesek

$$F_g = \mu \times N \quad (2.3)$$

Keterangan:

$F_g$  = Gaya Gesek (N)

$\mu$  = Koefisien gesekan

$N$  = Gaya normal (N)

#### 2. Gaya Berat

$$w = m \times g \quad (2.4)$$

Keterangan:

$w$  = Gaya Berat (N)

$m$  = Massa benda (kg)

$g$  = Gravitasi bumi ( $m/s^2$ )

#### 3. Gaya Sejenis

$$s = \rho \times g \text{ atau } s = \frac{w}{V} \quad (2.5)$$

Keterangan:

$s$  = Berat jenis ( $N/m^3$ )

$w$  = Berat benda (N)

$V$  = Volume benda ( $m^3$ )

$\rho$  = Massa jenis ( $kg/m^3$ )

## 2.9 Pesawat Sederhana (Tuas)[15]

Tuas atau pengungkit biasa kita gunakan dalam kehidupan sehari-hari. Contoh pada saat kita membuka tutup botol, jungkat-jungkit, linggis, dan lain sebagainya. Tuas digunakan untuk memindahkan benda yang berat.

Prinsipnya adalah, semakin jauh jarak kuasa terhadap titik tumpu, maka semakin kecil gaya yang diperlukan untuk mengangkat suatu beban. Rumusnya adalah sebagai berikut:

$$W \times L_b = F \times L_k \quad (2.6)$$

Keterangan:

$W$  = Beban (N)

$L_b$  = Jarak beban

$F$  = Kuasa (N)

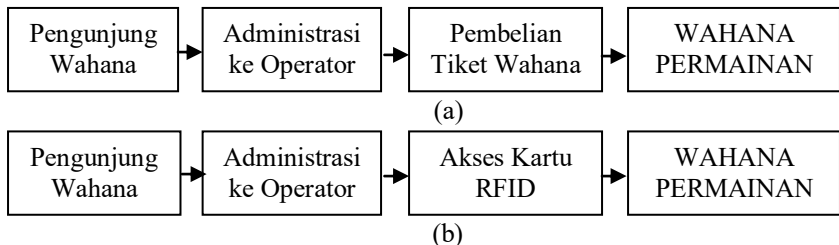
$L_k$  = Jarak kuasa

### BAB III

## PERANCANGAN SISTEM KONTROL

Pada bab ini akan dibahas mengenai perancangan dan pembuatan sistem pembayaran pasca bayar pada wahana taman bermain menggunakan RFID, baik perancangan perangkat keras (*hardware*), perancangan dan pembuatan perangkat lunak (*software*) serta perancangan mekanik.

Prinsip kerja dari sistem pembayaran menggunakan RFID pada wahana taman bermain ini awal mulanya masih terdapat kekurangan dalam melakukan transaksi dengan menggunakan metode lama, yaitu pembelian tiket memakai uang tunai. Hal tersebut dapat menyebabkan antrian yang panjang karena proses transaksi yang berjalan lambat, sehingga mengakibatkan antrian pengunjung menumpuk pada loket pembelian tiket wahana permainan. Disini akan dibahas terlebih dahulu tentang alur sistematika yang digunakan di wahana permainan,



**Gambar 1.1** Alur Sistematika pada Wahana Permainan

Pada alur sistematika gambar pertama, memiliki beberapa kekurangan dalam penerapan di wahana permainan seperti untuk dapat menggunakan wahana terlebih dahulu harus melakukan pembelian tiket dengan nominal yang telah ditentukan pada wahana permainan tersebut. Jika pengunjung melakukan pembayaran melebihi dari nominal yang ditentukan maka penjaga loket harus memberikan kembalian kepada pengunjung. Hal seperti ini dapat menyebabkan antrian yang bertambah akibat transaksi yang terjadi, terutama jika pihak penjaga loket kekurangan uang tunai saat memberikan kembalian.

Alur sistematika gambar kedua lebih mudah karena fungsi tiket digantikan dengan menggunakan kartu RFID sehingga tidak perlu lagi



melakukan transaksi oleh penjaga loket wahana permainan. Jadi ini lebih memudahkan untuk kedua belah pihak.

Untuk perancangan perangkat keras (*hardware*) meliputi:

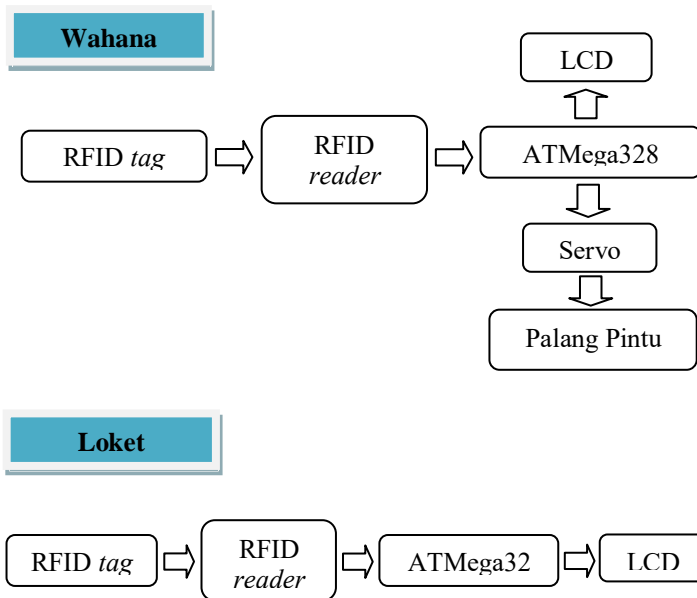
1. Modul *Regulator* LM2596
2. Rangkaian mikrokontroler ATmega328
3. Rangkaian *display* dengan LCD 16x2
4. Rangkaian RFID
5. Rangkaian Motor *Servo*
6. Rangkaian Komunikasi I2C
7. *Power Supply*

Sedangkan untuk pembuatan perangkat lunak (*software*) meliputi pembuatan program menggunakan bahasa pemrograman arduino yang nantinya akan disimpan dalam mikrokontroler ATmega328 untuk mengolah data masukan dan keluaran dan pembuatan program sistem informasi saldo dengan menggunakan pemrograman arduino. Masukan disini berupa RFID *tag* yang akan dibaca oleh RFID *reader* dan keluarannya berupa LCD, dan Motor *Servo*.

Sistem pembayaran ini mulai dari pembagian kartu berupa *tag* RFID pada saat pengunjung melakukan administrasi diloket masuk taman bermain. Kartu tersebut digunakan sebagai pengganti uang tunai saat melakukan transaksi pada wahana permainan. Dalam proses transaksi pembayaran di wahana permainan ini dilakukan dengan metode pasca bayar, dimana pengunjung menggunakan kartu tersebut untuk mengaktifkan palang pintu wahana dan pembayaran dilakukan pada saat pengunjung akan meninggalkan taman bermain. Transaksi pembayaran ini memudahkan pengunjung dalam mengakses wahana permainan karena proses masuk wahana yang cepat sehingga tidak terjadi lagi antrian pengunjung.

### 1.1 Perancangan Elektronika

Perancangan elektronika terdiri dari rangkaian *minimum system* mikrokontroler ATmega328, LCD, RFID, dan motor DC. Berikut ini merupakan blok fungsional dari sistem pembayaran pada wahana. Pada blok fungsional wahana, RFID *tag* di dekatkan kepada RFID *reader* yang telah terhubung dengan ATmega328 dan *output* akan ditampilkan pada LCD dan menggerakkan palang pintu wahana. Pada blok fungsional loket, RFID *tag* didekatkan kepada RFID *reader* yang telah terhubung dengan ATmega328 dan *output* akan ditampilkan pada LCD.



**Gambar 1.2** Diagram Fungsional Alat

### 1.1.1 Cara Kerja Sistem

Sistem pembayaran pada wahana permainan ini terhubung antara pintu pada tiap-tiap wahana dengan loket pembayaran taman bermain.

Sistem kerjanya pertama RFID tag akan dibaca oleh RFID reader, data yang dikirim akan diterima oleh mikrokontroler. Mikrokontroler yang menerima ID dari RFID tag jika cocok dengan program yang telah disimpan di mikrokontroler maka akan mengirimkan *command* (perintah) untuk mengaktifkan motor *servo* dimana kemudian palang pintu wahana akan terbuka. Secara waktu yang bersamaan mikrokontroler mengirimkan informasi karakter ke LCD.

#### 1. Mikrokontroler ATMega328

Disini kita menggunakan sebuah ATMega328, mikrokontroler ini berfungsi untuk menyimpan *database* dari RFID dan memproses *input* digital dari RFID apakah sesuai *database* yang telah di program sebelumnya atau tidak, jika data sesuai

maka mikrokontroler ini akan mengaktifkan motor *servo*. Mikrokontroler disini juga berfungsi sebagai pengatur kapan motor *servo* akan aktif.

2. **Sensor RFID**

RFID berfungsi sebagai indikator untuk mengaktifkan Motor *servo*. Terdiri dari RFID *reader* dan RFID *tag* yang digunakan sebagai identitas dari RFID tersebut.

3. **LCD 16x2**

LCD 16x2 berfungsi untuk menampilkan biaya yang dibutuhkan oleh pengunjung dan digunakan sebagai indikator bahwa pengunjung telah berhasil mengakses wahana permainan.

4. **Motor Servo**

Motor *Servo* yang digunakan untuk mengatur buka tutup palang pintu wahana.

5. **Power Supply**

*Power supply* disini berfungsi sebagai penyuplai tegangan listrik pada alat.

### 1.1.2 Modul *Regulator* LM2596

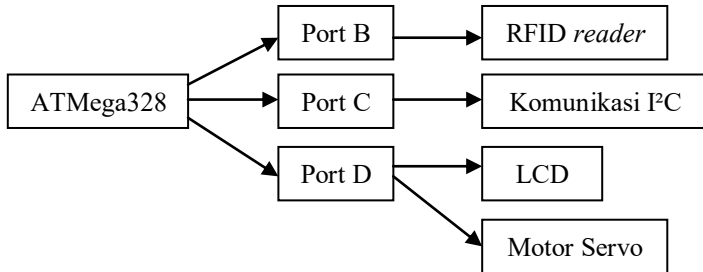
Modul *Regulator* disini sebagai modul DC DC *step down regulator* yang mengubah tingkatan tegangan (*voltage level*) arus searah menjadi lebih rendah isbanding tegangan masukan, digunakan untuk mengatur gerak motor *servo* pada alat. Tegangan *input* dapat dialiri tegangan antara 3V hingga 40V DC, yang diubah menjadi tegangan yang lebih rendah antara 1,5V hingga 35V DC. Pada modul *regulator* ini terdapat potensio yang dapat digunakan untuk mengatur tegangan keluaran.



**Gambar 1.3** Modul *Regulator* LM2596[16]

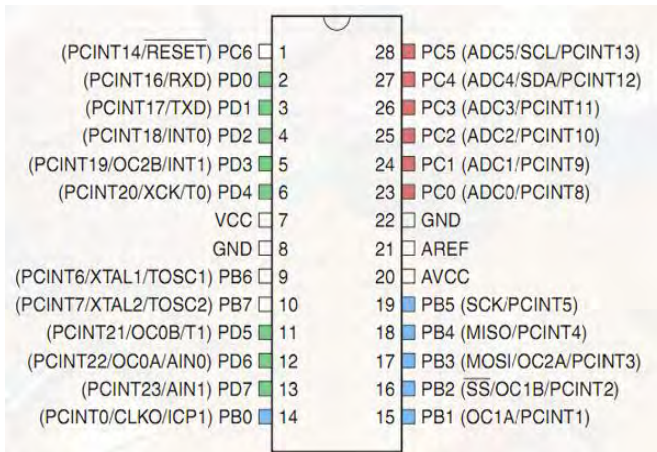
### 1.1.3 Rangkaian Mikrokontroler ATmega328[5]

Mikrokontroler ATmega328 adalah otak dari rangkaian untuk mengatur semua perintah eksekusi yang diberikan saat pengisian program.



**Gambar 1.4** Fungsional Mikrokontroler ATmega328

Pada Mikrokontroler ATmega328 ini memiliki beberapa fungsi utama yaitu untuk menerima dan mengolah data dari RFID tag untuk mengetahui biaya yang harus dikeluarkan pengunjung untuk memasuki wahana permainan dan ditampilkan pada LCD, kemudian menggerakkan motor servo sebagai portal pintu wahana sebagai tanda bahwa pengunjung telah diperbolehkan untuk memasuki wahana permainan.



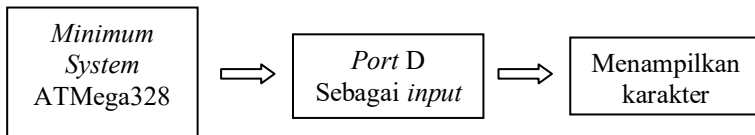
**Gambar 1.5** Port I/O ATmega328[17]

Fungsi pada tiap-tiap *port* Mikrokontroler ATmega328 yang terhubung dengan alat adalah sebagai berikut:

1. **Port B** dianalogikan untuk mengolah data pembacaan data dari RFID *reader*.
2. **Port C** dianalogikan untuk komunikasi I2C pada sistem pembayaran.
3. **Port D** dianalogikan untuk menampilkan perintah pada LCD dan Motor *Servo*

#### 1.1.4 Liquid Cristal Display (LCD)

LCD berfungsi untuk menampilkan hasil akuisisi data mikrokontroler. LCD tersebut digunakan untuk menginformasikan dan menandai bahwa total biaya yang dibutuhkan pengunjung wahana telah terdeteksi dan siap untuk memasuki wahana permainan.



**Gambar 1.6** Fungsional LCD

Data yang telah diolah oleh mikrokontroler akan dikirimkan oleh *Port D* untuk menampilkan karakter kata yang memudahkan dalam pemahaman *user* pada saat melakukan transaksi untuk memasuki wahana permainan.

Untuk menghubungkan LCD 16x2 ke mikrokontroler dapat langsung menyesuaikan instruksi dari program arduino dengan menyesuaikan *Port* yang akan dihubungkan dari mikrokontroler ke LCD.

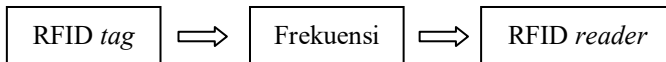
*Wiring* Mikrokontroler ATmega328 dan LCD adalah sebagai berikut:

1. *Pin D00* ATmega328 to *Port D7* LCD
2. *Pin D01* ATmega328 to *Port D6* LCD
3. *Pin D02* ATmega328 to *Port D5* LCD
4. *Pin D03* ATmega328 to *Port D4* LCD
5. *Pin D04* ATmega328 to *Port Enable* LCD
6. *Pin D05* ATmega328 to *Port Rs* LCD
7. *Pin Rw* LCD to *Ground*

### 1.1.5 Rangkaian RFID

RFID *reader* yang digunakan pada tugas akhir ini adalah RC522. Sedangkan kartu *tag* RFID digunakan pada tugas akhir ini adalah MIFARE *card*. Dalam *tag* RFID ini terdapat informasi yang digunakan untuk menyimpan informasi berupa identitas dari *tag* itu sendiri.

Apabila *tag* (kartu) didekatkan pada RFID *reader*, maka mikrokontroler akan mencocokkan apakah RFID *tag* yang dibaca sesuai dengan *database* yang telah di *input* atau tidak. Jika *tag* sesuai maka motor *servo* akan aktif dan LCD akan menampilkan total biaya pengunjung, namun jika tidak sesuai maka LCD akan menampilkan bahwa *tag* pengunjung tidak teregistrasi dan motor *servo* tidak aktif.



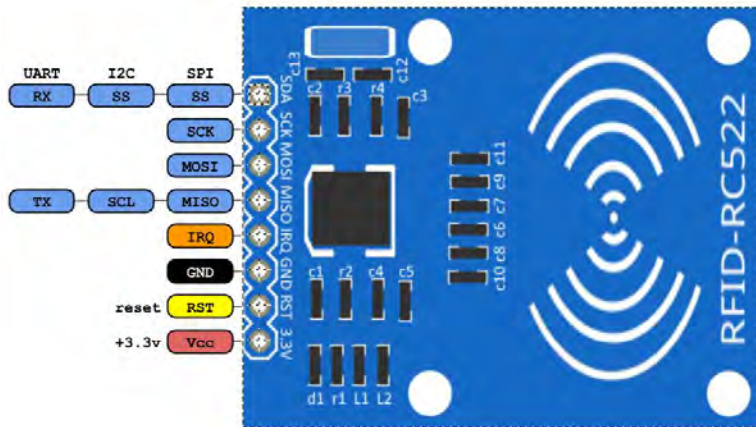
**Gambar 1.7** Fungsional RFID

RFID *tag* memiliki kode *serial number* yang memiliki karakteristik berbeda-beda. Kode *serial* dibaca oleh RFID *reader* yang selanjutnya akan diolah kembali oleh mikrokontroler.

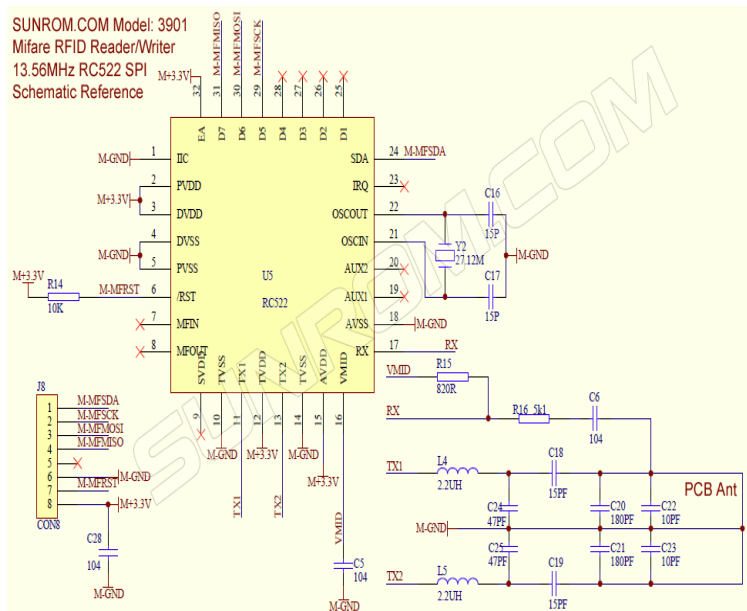


**Gambar 1.8** RFID Tag[18]

Untuk fungsi tiap-tiap *pin* dan *output* data format dari RFID RC522 ditunjukkan pada Gambar 3.9

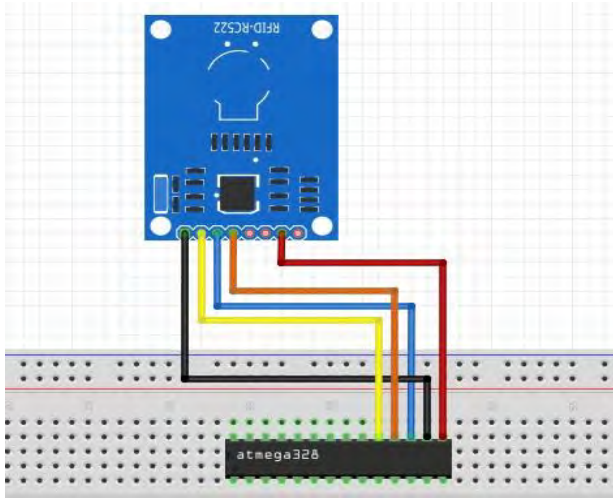


Gambar 1.9 RFID RC522 Pin Out[19]



Gambar 1.10 Skematik RFID Reader RC522[20]

Sedangkan untuk gambaran rangkaian RFID yang digunakan dapat dilihat pada Gambar 3.11



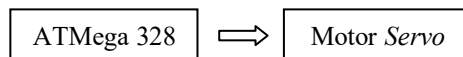
**Gambar 1.11** Rangkaian RFID

*Wiring* Mikrokontroler ATmega328 dan RFID RC522 pada alat adalah sebagai berikut:

1. *Pin* B01 ATmega328 to *Port* RSI RFID RC522
2. *Pin* B02 ATmega328 to *Port* SS RFID RC522
3. *Pin* B03 ATmega328 to *Port* MOSI RFID RC522
4. *Pin* B04 ATmega328 to *Port* MISO RFID RC522
5. *Pin* B05 ATmega328 to *Port* RSI RFID RC522

### 1.1.6 Rangkaian Motor *Servo*

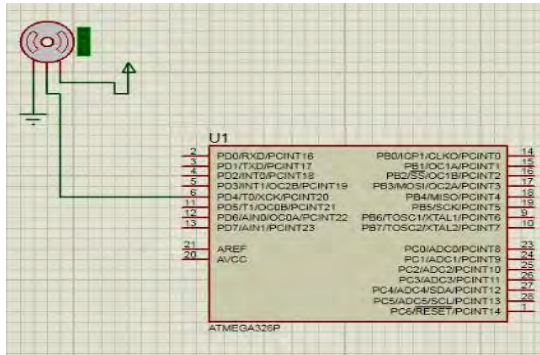
Pada alat ini menggunakan Motor *Servo* dengan torsi maksimal seberat 4kg. Motor *Servo* digunakan pada alat sebagai portal pintu wahana permainan. Pada Gambar 3.13 digambarkan bagan fungsional dari motor *servo*.



**Gambar 1.12** Fungsional Motor *Servo*



Skematik pada rangkaian Motor *Servo* yang digunakan pada alat adalah pada Gambar 3.13



**Gambar 1.13** Rangkaian Motor *Servo* dengan *Proteus*

*Wiring* Mikrokontroler ATMEGA328 dan Motor *Servo* pada alat adalah sebagai berikut:

1. *Pin* D04 ATMEGA328 to Motor *Servo*

Pada prinsip kerja motor *servo*, untuk menentukan lebar pulsa sebagai penentu posisi sudut *servo* yaitu dengan mengendalikannya menggunakan PWM melalui kabel kontrol. Pada alat ini ketika pintu wahana menandakan terbuka, motor *servo* mengangkat palang pintu sekitar 90°, jadi lebar pulsa harus 1,71ms seperti yang telah ditentukan. Saat palang menandakan tutup, alat menggunakan sudut sebesar 10°. Jadi lebar pulsa yang dibutuhkan adalah sebagai berikut:

$$\begin{aligned}
 90^\circ &= 1,71 \\
 0^\circ &= 1,52 \\
 \Delta 90^\circ - 0^\circ &= 0,19 \\
 \frac{\Delta 90^\circ - 0^\circ}{9} &= \frac{0,19}{x} \\
 \frac{1}{9} &= \frac{x}{0,19} \\
 x &= \frac{0,19}{9} \\
 &= 0,02
 \end{aligned}
 \tag{3.1}$$

Jadi setiap akan mengubah 10° pada palang pintu wahana, maka lebar sinyal harus diubah menjadi 0,02ms.



## 1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak sistem pembayaran pada wahana taman bermain ini dengan menggunakan aplikasi arduino *Integrated Development Environment* (IDE). Program digunakan untuk mengisi mikrokontroler dengan bahasa yang digunakan adalah bahasa pemrograman arduino.

### 1.2.1 Aplikasi Arduino *Integrated Development Environment* (IDE)

IC ATmega328 yang terdapat pada mikrokontroler akan mengolah data untuk memberi perintah kepada perangkat lain dalam suatu sistem. Untuk dapat menjalankan urutan proses kerja sistem yang diinginkan, maka dibutuhkan aplikasi arduino *Integrated Development Environment* (IDE).

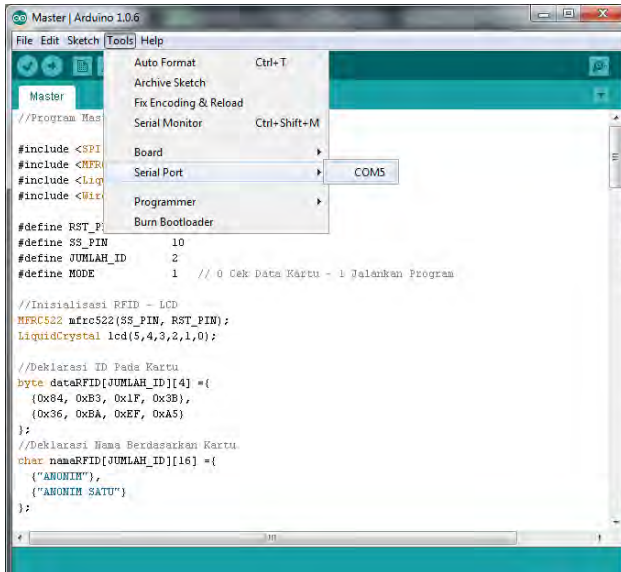
Aplikasi Arduino *Integrated Development Environment* (IDE) berfungsi sebagai sarana dalam menuliskan perintah kerja yang akan diunggah ke dalam mikrokontroler ATmega328. Selain menuliskan kode program, Arduino *Integrated Development Environment* (IDE) juga dapat mengkonversi kode program berupa bahasa manusia ke dalam bentuk bahasa mesin.

Dalam perancangan program pada *software* arduino *Integrated Development Environment* (IDE) dengan fungsi terkait diperlukan beberapa tahapan yang harus dilakukan terlebih dahulu. Tahapan tersebut adalah membuat algoritma dari alat yang sudah kita jalankan. Setelah itu barulah kita memprogram fungsi terkait.

Untuk melakukan pemrograman dilakukan beberapa tahapan:

1. **Setting Board.** Dalam pemrograman *software* arduino *Integrated Development Environment* (IDE) harus di *setting* terlebih dahulu *board* agar cocok dengan penggunaan. Dalam alat kali ini mikrokontroler menggunakan IC ATmega328 yang merupakan IC Arduino UNO R3. Untuk *setting board* bisa masuk ke *tools – board* – setelah itu pilihlah *board* yang sesuai.
2. **Setting Serial.** *Serial* ini merupakan kabel dari mikrokontroler ATmega328 yang dihubungkan kepada komputer atau laptop. *Setting serial* bisa masuk *tools - serial* - lalu pilih COM yang sesuai dengan mikrokontroler yang terpasang. Untuk lebih jelasnya dapat dilihat pada Gambar 3.15
3. Apabila *program* tidak dapat di *download* karena *serial port*, maka cek terlebih dahulu *serial* yang benar pada *device manager*. Lalu dalam *software* arduino *Integrated Development Environment*

Environment (IDE) untuk memilih serial port nya samakan dengan serial port untuk mikrokontroler dalam device manager tersebut. Untuk masuk ke device manager dapat masuk start windows – lalu ketika device manager klik dua kali dan masuk ke COM.

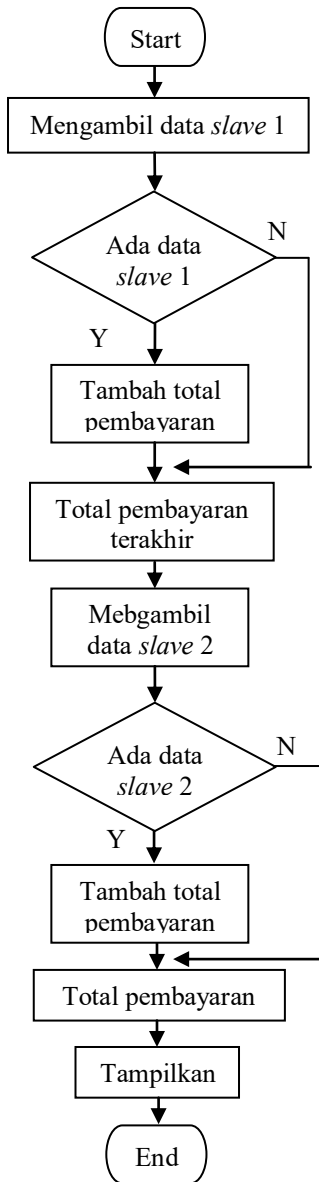


**Gambar 1.15** Setting Serial Port

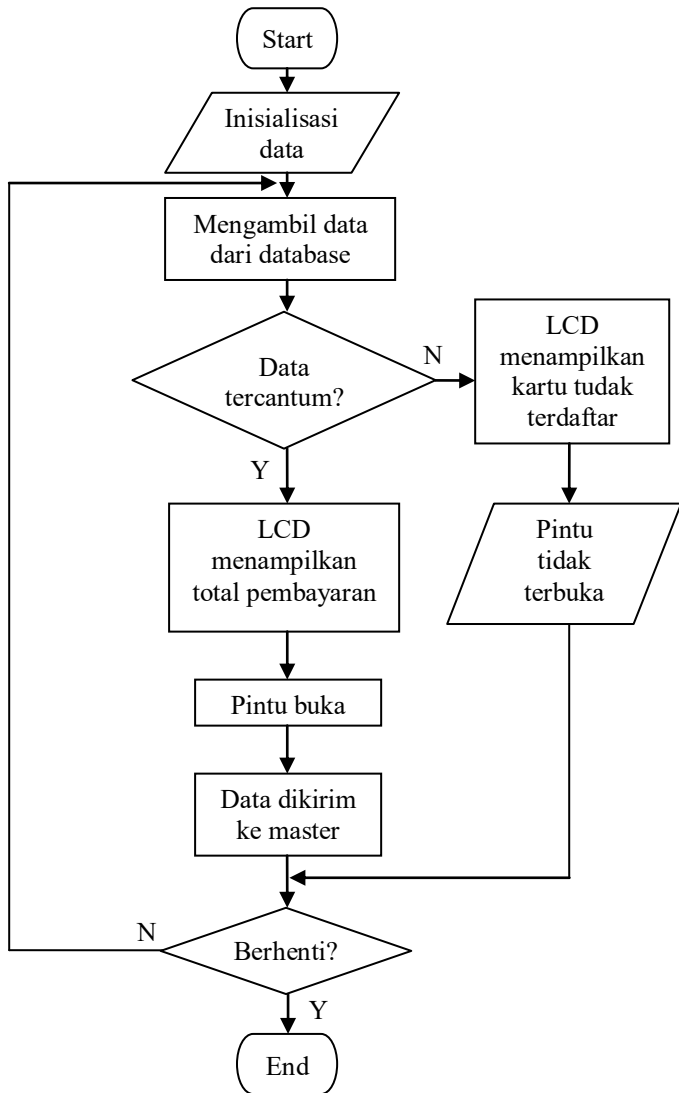
### 1.2.2 Pembuatan Flowchart Program

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya.

Pada pembuatan program flowchart pembayaran wahana taman bermain ini dibagi menjadi dua yaitu master dan slave. Program master diletakkan pada loket pembayaran taman bermain sebagai penjumlah dari total transaksi pengunjung, sedangkan program slave diletakkan pada tiap-tiap wahana permainan sebagai pendeteksi biaya transaksi yang akan dikeluarkan oleh pengunjung. Berikut adalah flowchart program dari alat ini.



**Gambar 1.16** Flowchart Master Sistem Wahana



**Gambar 1.17** Flowchart Slave Sistem Wahana

### **1.2.3 Perancangan Perangkat Lunak untuk RFID**

RFID pada alat ini bertindak sebagai *receiver*. RFID *reader* membaca ID dari RFID *tag* yang terdapat pada kartu pengunjung wahana. Algoritma pembuatan program untuk RFID ini adalah sebagai berikut :

1. RFID dalam kondisi aktif dan siap untuk menerima data.
2. Ketika *tag* didekatkan ke *reader*, RFID akan membaca data tersebut.
3. Setelah membaca kode-kode dari *tag*, maka akan diteruskan ke mikrokontroler untuk diproses apakah kode tersebut sesuai atau tidak.
4. LCD akan menampilkan biaya pada wahana jika ID yang dibaca oleh RFID *reader* sesuai dan kemudian mengaktifkan motor *servo* sebagai portal pintu wahana permainan.

### **1.2.4 Perancangan Software Pengelola Database**

Dalam perancangan perangkat lunak untuk mengelola *database* digunakan Arduino *Integrated Development Environment* (IDE) yang telah memiliki fasilitas-fasilitas yang lengkap dan mudah untuk mengelola *database*.

Pembuatan program untuk pengelola *database* sistem pembayaran pasca bayar pada wahana permainan adalah sebagai berikut:

1. Pembuatan program *slave* yang diletakkan pada tiap-tiap wahana permainan, fungsinya sebagai *form* informasi saldo yang dikenakan kepada pengunjung yang akan menikmati wahana permainan.
2. Pembuatan program *master* yang diletakkan di loket pembayaran taman bermain, fungsinya sebagai *form* pembayaran pengunjung ketika telah selesai menaiki wahana permainan dan akan meninggalkan taman bermain.

### **1.3 Perancangan Mekanik**

Mekanik dirancang dan dibuat untuk menunjang kelengkapan dari Tugas Akhir. Mekanik yang dibuat mempunyai struktur dan fungsi yang berbeda-beda.

### 1.3.1 Pintu Wahana

Pintu wahana pada Tugas Akhir ini dibuat dengan menggunakan bahan dasar aluminium. Pintu Wahana ini memiliki tinggi 2 meter, lebar 100cm, dan ketebalan pintu 8cm.



**Gambar 1.18** Desain Pintu Wahana

### 1.3.2 Palang Pintu

Palang pada alat ini berfungsi sebagai tanda diperbolehkannya pengunjung memasuki wahana permainan. Palang ini memiliki masa beban 0,25kg. Panjang keseluruhan palang adalah 120cm, dengan panjang lengan masa 20cm dan panjang lengan beban 100cm.

Langkah-langkah dalam penambahan berat pada lengan kuasa dirumuskan sebagai berikut:

1. Menggunakan rumus tuas

Diketahui : Masa beban ( $M_b$ ) 125 g = 0,125 kg  
 $Gravity$  ( $g$ ) = 10  
Lengan beban ( $L_b$ ) 100 cm = 1 m  
Lengan kuasa ( $L_k$ ) 20 cm = 0,2 m  
Masa kuasa ( $M_k$ ) 25 g = 0,025 kg

Ditanya :  $F$ ?

Jawab :  $W \times L_b = F \times L_k$  (3.2)

$$(M_b \times g) \times L_b = F \times L_k$$

$$(0,125 \times 10) \times 1 = F \times 0,2$$

$$\frac{1,25 \times 1}{0,2} = F$$

$$\frac{1,25}{0,2} = F$$

$$6,25 = F$$



## 2. Menggunakan Hukum Newton 1

Diketahui : F beban = 6,25  
F kuasa = 3  
Gravity = 10

Ditanya :  $mx$ ?

Jawab :  $Fb - Fk = Fx = 0$  (3.3)

$$6,25 - 3 = Fx = 0$$

$$3,25 = Fx = 0$$

$$3,25 = ((mx) \times g) = 0$$

$$3,25 = ((mx) \times 10) = 0$$

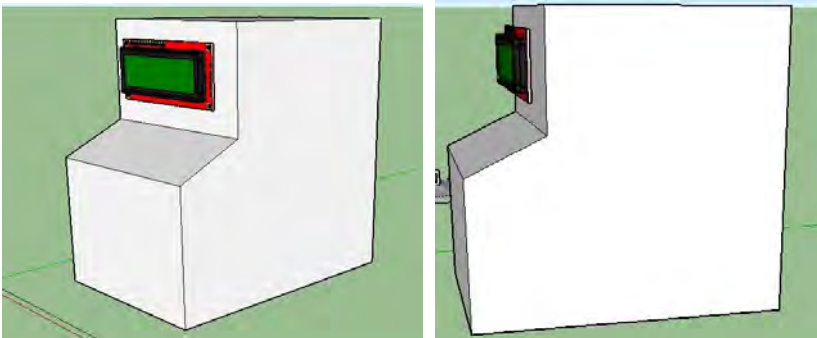
$$mx = \frac{3,25}{10} = 0$$

$$mx = 0,325 \text{ kg}$$

Jadi penambahan berat yang dibutuhkan pada lengan kuasa adalah 0,325kg.

### 1.3.3 Packaging Rangkaian

*Packaging* Rangkaian pada Tugas Akhir ini dibuat dengan menggunakan bahan dasar *acrylic*. *Packaging* diciptakan sebagai wadah untuk mikrokontroler ATmega328, LCD, dan RFID reader. Letak *packaging* rangkaian ini berada di samping pintu wahana permainan. Untuk peletakan LCD pada *packaging* rangkaian berada di bagian atas dari tempat peletakan RFID reader. Sedangkan untuk peletakan mikrokontroler ATmega328 diletakkan tertata pada bagian dalam *packaging* rangkaian.



**Gambar 1.19** Desain *Packaging* Rangkaian

## BAB IV PENGUJIAN DAN ANALISA DATA

Pengujian merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perancangan dapat dilihat dari hasil-hasil yang dicapai selama pengujian sistem. Pengujian juga bertujuan untuk mengetahui kelebihan dan kekurangan dari sistem yang telah dibuat. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan pada sistem. Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini sesuai dengan blok fungsional alat pada Gambar 3.2 telah terlaksana atau tidak, maka perlu dilakukan pengujian dan analisa terhadap alat yang dibuat.

### 2.1 Pengujian pada RFID RC522

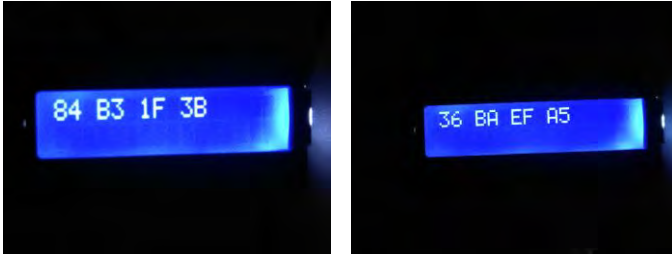
RFID adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah piranti kecil yang disebut *tag* atau transponder. *Tag* RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari piranti yang kompatibel, yaitu RFID *reader* dengan jarak kisaran pembacaan 4cm, serta bekerja pada frekuensi 13,56MHz.

Kami melakukan pengujian pada RFID *reader* RC522 dengan cara menghubungkan RFID *reader* dengan rangkaian mikrokontroler ATmega328. Untuk menampilkan data yang ada pada RFID *tag* dengan cara menuliskan perintah menggunakan *software* Arduino *Integrated Development Environment* (IDE) dan di *upload* pada mikrokontroler ATmega328. *Input* kemudian akan ditampilkan pada LCD.

```
//Fungsi yang digunakan untuk melihat data yang ada pada kartu
void rfidData(void){
    lcd.clear();
    for(byte i = 0; i < mfrc522.uid.size; i++) {
        lcd.print(mfrc522.uid.uidByte[i], HEX);
        lcd.print(' ');
    }
    delay(5000);
}
```

**Gambar 2.1** Program untuk Cek Data Kartu

Saat RFID *tag* didekatkan pada RFID *reader* RC522 pada jarak dan kemiringan tertentu maka tampilan pada LCD akan muncul ID yang terdapat pada RFID *tag* tersebut.



**Gambar 2.2** ID pada RFID *Tag*

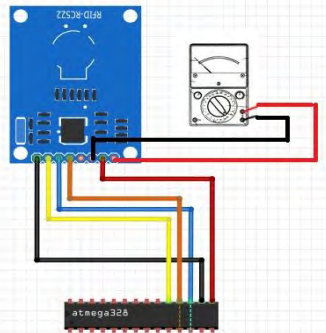
Perangkat RFID (RFID *tag* dan RFID *reader*) akan bekerja apabila frekuensinya sama. Ada 4 macam frekuensi kerja RFID, yaitu:

- *Low Frequency* (LF) : 125-134 KHz
- *High Frequency* (HF) : 13,56 MHz
- *Ultra High Frequency* (UHF) : 868-956 MHz
- *Microwave* : 2,45 GHz

Adapun RFID *reader* RC522 tergolong pada frekuensi *High Frequency* (HF) dengan spesifikasi sebagai berikut:

- *Working current* : 13-26mA/ DC 3,3V
- *Standby current* : 10-13mA/ DC 3,3V
- *Sleeping current* : <80uA
- *Peak current* : <30mA
- Frekuensi kerja : 13,56 MHz
- Jarak pembacaan : 0-4cm
- Protocol : SPI
- Kecepatan komunikasi data hingga 10Mbit/s
- *Max SPI speed* : 10Mbit/s

Dalam pengujian RFID ini, kami membandingkan jarak antara *tag* dan *reader* dari RFID, sehingga dapat diketahui jarak berapakah RFID ini dapat bekerja dan mencari jarak dan posisi paling tepat dalam pembacaan RFID menggunakan RC522. Kami melakukan pengujian dengan cara mengukur jarak pada masing-masing RFID *tag* yang didekatkan pada RFID *reader*, dimulai dari jarak 0 cm sampai dengan RFID *tag* tidak dapat terdeteksi oleh RFID *reader*. Cara pengukuran ialah dengan menggunakan alat ukur penggaris. Untuk pengukuran tegangan pada RFID, *input* VCC pada RFID dihubungkan dengan *probe positive* pada multimeter dan *ground* pada RFID dihubungkan dengan *probe negative* pada multimeter.



**Gambar 2.3** Pengukuran RFID menggunakan Multimeter

Pada pengukuran ini kami mengukur lima kartu *tag* RFID yaitu kartu *tag* dengan kode 5A 7E F6 A5, kartu *tag* dengan kode B0 64 70 B5, kartu *tag* dengan kode 36 BA EF A5, kartu *tag* dengan kode 84 B3 1F 3B, dan kartu *tag* dengan kode E0 50 C2 1E.

Hasil pengukuran pada kartu *tag* dengan kode 5A 7E F6 A5 ditampilkan pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3. Dari hasil pengukuran tersebut didapatkan perbandingan jarak pada masing-masing tabel dari kartu *tag* dengan kode 5A 7E F6 A5 seperti pada Gambar 4.4.

**Tabel 2.1** RFID *Reader* pada Wahana 1

Kode RFID <i>tag</i>	Jarak (cm)	Tegangan <i>Input</i> (V)	Keterangan
5A 7E F6 A5	0	3,82	Terdeteksi
	1	3,81	Terdeteksi

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
5A 7E F6 A5	1,5	3,79	Terdeteksi
	2	3,79	Terdeteksi
	2,5	3,78	Terdeteksi
	3	3,78	Terdeteksi
	3,5	3,72	Terdeteksi
	3,7	3,70	Tidak Terdeteksi
	3,9	3,69	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

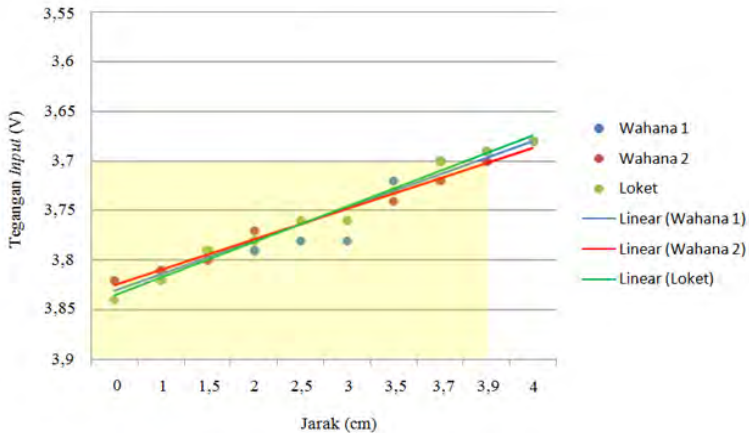
**Tabel 2.2** RFID Reader pada Wahana 2

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
5A 7E F6 A5	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,80	Terdeteksi
	2	3,77	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,74	Terdeteksi
	3,7	3,72	Terdeteksi
	3,9	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.3** RFID Reader pada Loket Pembayaran

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
5A 7E F6 A5	0	3,84	Terdeteksi
	1	3,82	Terdeteksi
	1,5	3,79	Terdeteksi
	2	3,78	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,73	Terdeteksi
	3,7	3,70	Tidak Terdeteksi
	3,9	3,69	Tidak Terdeteksi

Kode RFID tag	Jarak (cm)	Tegangan Input (V)	Keterangan
5A 7E F6 A5	4	3,68	Tidak Terdeteksi



**Gambar 2.4** Perbandingan Jarak RFID Tag 5A 7E F6 A5

- Garis biru pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,016x - 3,848$
- Garis merah pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,015x - 3,84$
- Garis hijau pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,017x - 3,853$

Hasil pengukuran pada kartu tag dengan kode B0 64 70 B5 ditampilkan pada Tabel 4.4, Tabel 4.5, dan Tabel 4.6. Dari hasil pengukuran tersebut didapatkan perbandingan jarak pada masing-masing tabel dari kartu tag dengan kode B0 64 70 B5 seperti pada Gambar 4.5.

**Tabel 2.4** RFID Reader pada Wahana 1

Kode RFID tag	Jarak (cm)	Tegangan Input (V)	Keterangan
B0 64 70 B5	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,79	Terdeteksi

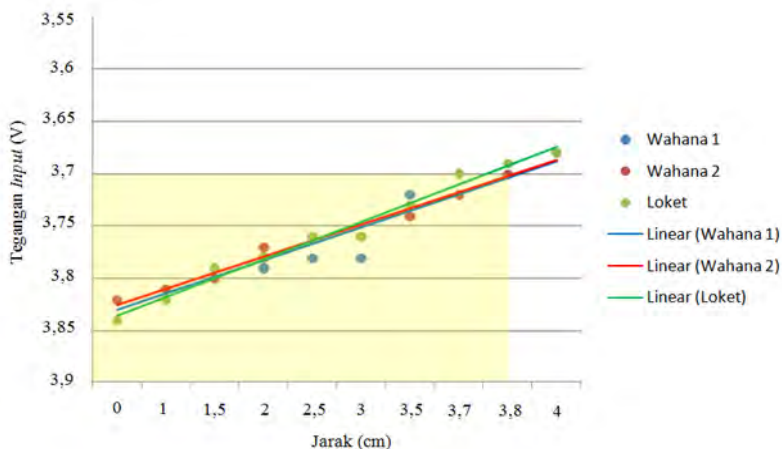
<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
B0 64 70 B5	2	3,79	Terdeteksi
	2,5	3,78	Terdeteksi
	3	3,78	Terdeteksi
	3,5	3,72	Terdeteksi
	3,7	3,72	Terdeteksi
	3,8	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.5** RFID *Reader* pada Wahana 2

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
B0 64 70 B5	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,80	Terdeteksi
	2	3,77	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,74	Terdeteksi
	3,7	3,72	Terdeteksi
	3,8	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.6** RFID *Reader* pada Loket Pembayaran

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
B0 64 70 B5	0	3,84	Terdeteksi
	1	3,82	Terdeteksi
	1,5	3,79	Terdeteksi
	2	3,78	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,73	Terdeteksi
	3,7	3,70	Tidak Terdeteksi
	3,8	3,69	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi



**Gambar 2.5** Perbandingan Jarak RFID Tag B0 64 70 B5

- Garis biru pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,015x + 3,845$
- Garis merah pada grafik di atas memiliki persamaan *linear* yaitu  $y = -0,015x + 3,84$
- Garis hijau pada grafik di atas memiliki persamaan *linear* yaitu  $y = -0,017x + 3,853$

Hasil pengukuran pada kartu tag dengan kode 36 BA EF A5 ditampilkan pada Tabel 4.7, Tabel 4.8, dan Tabel 4.9. Dari hasil pengukuran tersebut didapatkan perbandingan jarak pada masing-masing tabel dari kartu tag dengan kode 36 BA EF A5 seperti pada Gambar 4.6.

**Tabel 2.7** RFID Reader pada Wahana 1

Kode RFID tag	Jarak (cm)	Tegangan Input (V)	Keterangan
36 BA EF A5	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,79	Terdeteksi
	2	3,79	Terdeteksi
	2,5	3,78	Terdeteksi
	3	3,78	Terdeteksi



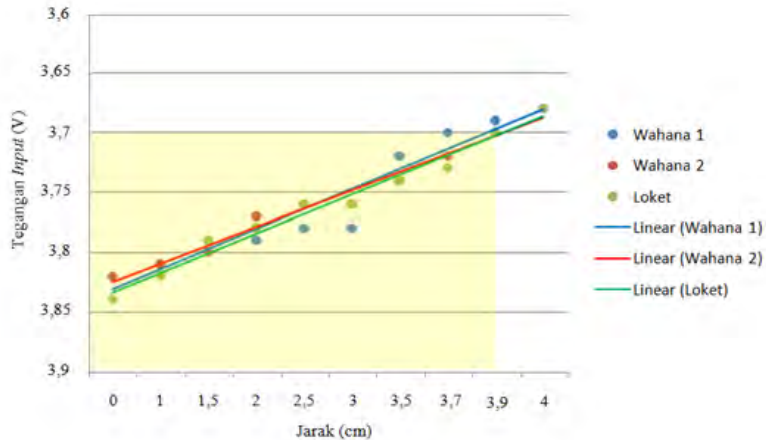
<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
36 BA EF A5	3,5	3,72	Terdeteksi
	3,7	3,70	Tidak Terdeteksi
	3,9	3,69	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.8** RFID *Reader* pada Wahana 2

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
36 BA EF A5	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,80	Terdeteksi
	2	3,77	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,74	Terdeteksi
	3,7	3,72	Terdeteksi
	3,9	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.9** RFID *Reader* pada Loket Pembayaran

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
36 BA EF A5	0	3,84	Terdeteksi
	1	3,82	Terdeteksi
	1,5	3,79	Terdeteksi
	2	3,78	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,74	Terdeteksi
	3,7	3,73	Terdeteksi
	3,9	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi



**Gambar 2.6** Perbandingan Jarak RFID *Tag* B0 64 70 B5

- Garis biru pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,016x - 3,848$
- Garis merah pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,015x - 3,84$
- Garis hijau pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,016x - 3,85$

Hasil pengukuran pada kartu *tag* dengan kode 84 B3 1F 3B ditampilkan pada Tabel 4.10, Tabel 4.11, dan Tabel 4.12. Dari hasil pengukuran tersebut didapatkan perbandingan jarak pada masing-masing tabel dari kartu *tag* dengan kode 84 B3 1F 3B seperti pada Gambar 4.7.

**Tabel 2.10** RFID *Reader* pada Wahana 1

Kode RFID <i>tag</i>	Jarak (cm)	Tegangan <i>Input</i> (V)	Keterangan
84 B3 1F 3B	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,79	Terdeteksi
	2	3,79	Terdeteksi
	2,5	3,78	Terdeteksi
	3	3,78	Terdeteksi

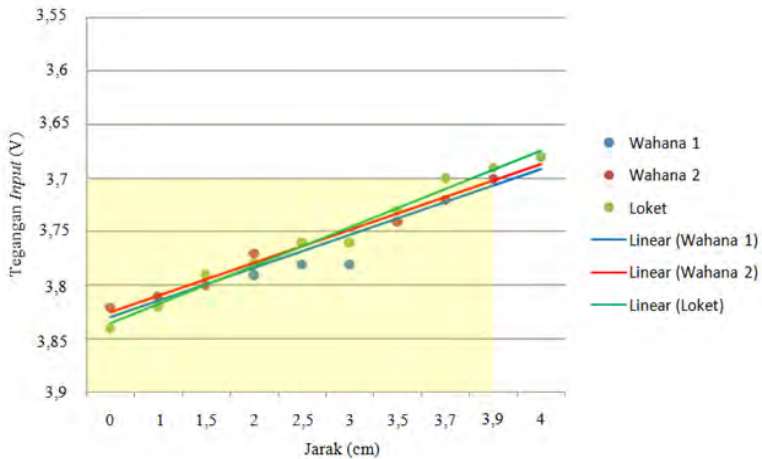
<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
84 B3 1F 3B	3,5	3,74	Terdeteksi
	3,7	3,72	Terdeteksi
	3,9	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.11** RFID Reader pada Wahana 2

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
84 B3 1F 3B	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,80	Terdeteksi
	2	3,77	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,74	Terdeteksi
	3,7	3,72	Terdeteksi
	3,9	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.12** RFID Reader pada Loket Pembayaran

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
84 B3 1F 3B	0	3,84	Terdeteksi
	1	3,82	Terdeteksi
	1,5	3,79	Terdeteksi
	2	3,78	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,73	Terdeteksi
	3,7	3,70	Tidak Terdeteksi
	3,9	3,69	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi



**Gambar 2.7** Perbandingan Jarak RFID *Tag* 84 B3 1F 3B

- Garis biru pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,015x - 3,845$
- Garis merah pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,015x - 3,84$
- Garis hijau pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,017x - 3,853$

Hasil pengukuran pada kartu *tag* dengan kode E0 50 C2 1E ditampilkan pada Tabel 4.13, Tabel 4.14, dan Tabel 4.15. Dari hasil pengukuran tersebut didapatkan perbandingan jarak pada masing-masing tabel dari kartu *tag* dengan kode E0 50 C2 1E seperti pada Gambar 4.8.

**Tabel 2.13** RFID *Reader* pada Wahana 1

Kode RFID <i>tag</i>	Jarak (cm)	Tegangan <i>Input</i> (V)	Keterangan
E0 50 C2 1E	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,79	Terdeteksi
	2	3,79	Terdeteksi
	2,5	3,78	Terdeteksi
	3	3,78	Terdeteksi

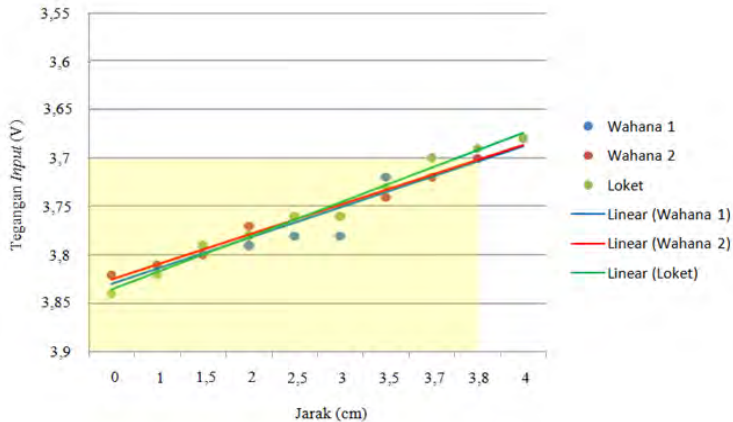
<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
E0 50 C2 1E	3,5	3,72	Terdeteksi
	3,7	3,72	Terdeteksi
	3,8	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.14** RFID Reader pada Wahana 2

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
E0 50 C2 1E	0	3,82	Terdeteksi
	1	3,81	Terdeteksi
	1,5	3,80	Terdeteksi
	2	3,77	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,74	Terdeteksi
	3,7	3,72	Terdeteksi
	3,8	3,70	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi

**Tabel 2.15** RFID Reader pada Loket Pembayaran

<b>Kode RFID tag</b>	<b>Jarak (cm)</b>	<b>Tegangan Input (V)</b>	<b>Keterangan</b>
E0 50 C2 1E	0	3,84	Terdeteksi
	1	3,82	Terdeteksi
	1,5	3,79	Terdeteksi
	2	3,78	Terdeteksi
	2,5	3,76	Terdeteksi
	3	3,76	Terdeteksi
	3,5	3,73	Terdeteksi
	3,7	3,70	Tidak Terdeteksi
	3,8	3,69	Tidak Terdeteksi
	4	3,68	Tidak Terdeteksi



**Gambar 2.8** Perbandingan Jarak RFID Tag E0 50 C2 1E

- Garis biru pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,015x - 3,845$
- Garis merah pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,015x - 3,84$
- Garis hijau pada grafik di atas memiliki persamaan *linear* yaitu  $y = 0,017x - 3,853$

Dari hasil pengukuran jarak pembacaan tag RFID pada RFID reader dapat disimpulkan bahwa tag RFID masih dapat dibaca meskipun tertutup oleh suatu benda kecuali oleh benda yang bersifat konduktor. Pada pengujian ini kami juga melakukan pengukuran tegangan dan arus pada RFID reader saat tanpa kartu tag RFID dan juga pada saat kartu tag RFID didekatkan.

## 2.2 Pengujian LCD

LCD merupakan alat untuk menampilkan karakter-karakter tertentu sesuai dengan program yang terdapat pada mikrokontroler ATmega328. Proses pengujian LCD pada alat ini dengan menggunakan *minimum system* mikrokontroler ATmega328. Pengujian ini menampilkan beberapa karakter di LCD yang menandakan bahwa LCD dan mikrokontroler dalam keadaan baik. Untuk melakukan pengujian terhadap LCD yaitu dengan cara menuliskan contoh perintah menggunakan *software* Arduino *Integrated Development Environment*

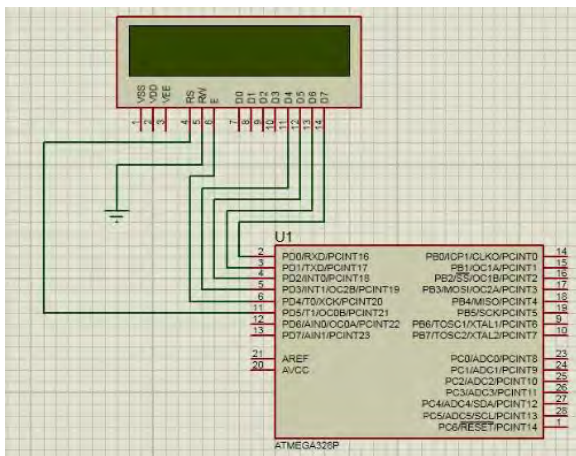
(IDE) dan di *upload* pada mikrokontroler ATmega328. *Input* kemudian akan ditampilkan pada LCD.

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(5, 4, 3, 2, 1, 6);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}
```

**Gambar 2.9** Program Perintah pada LCD



**Gambar 2.10** Rangkaian LCD dengan *Proteus*

### 2.3 Pengujian Keseluruhan

Pengujian alat secara keseluruhan bertujuan untuk mengetahui alat dapat bekerja dengan maksimal sesuai fungsinya. Pengujian ini juga dimaksudkan untuk mengetahui jika terdapat *error* pada alat. Dalam proses pengujian sistem ini dimulai dengan pada saat pengunjung memasuki area wahana permainan.

Pengunjung mendekatkan kartu RFID *tag* pada RFID *reader* untuk melakukan transaksi. Kartu RFID *tag* akan teridentifikasi oleh mikrokontroler jika ID pada RFID *tag* telah terdaftar dan akan tampil pada LCD "Teridentifikasi". Akan tetapi jika ID belum terdaftar pada mikrokontroler maka LCD akan menampilkan "Tidak Terdaftar".

```
if(foundID){
    lcd.print("Teridentifikasi");
    delay(500);
    lcd.clear();
    lcd.setCursor(0,0);
}else{
    lcd.print(" Tidak Terdaftar");
}
```

**Gambar 2.11** Program untuk Mendeteksi *Tag* dan Ditampilkan di LCD

Selanjutnya jika transaksi telah selesai dilakukan, palang pintu wahana akan terbuka. Motor *servo* akan menggerakkan palang pintu sehingga posisi palang 90°, kemudian setelah beberapa detik palang pintu kembali ke posisi semula 0°.

```
void gate(void){
    myServo.write(20); //tutup
    delay(2000); //bekerjapada 2ms
    myServo.write(90); //buka
    delay(2000);
    myServo.write(20); //tutup
    delay(2000);
}
```

**Gambar 2.12** Program Motor *Servo*

Pada proses akhir pengunjung melakukan total pembayaran pada loket keluar. Mikrokontroler pada loket pembayaran terhubung dengan mikrokontroler pada setiap wahana menggunakan komunikasi I2C dan berperan sebagai *master*. Fungsi *master* adalah menjumlahkan total biaya pada masing-masing *slave* sehingga total pembayaran yang dibutuhkan dapat diketahui oleh pengunjung wahana.



-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN A

### A.1. Adapter 1 pada Pintu Wahana



### A.2. Adapter 2 pada Pintu Wahana



### A.3. Tampilan LCD pada Wahana



### A.4. RFID tag Teridentifikasi



### A.5. RFID tag Tidak Terdaftar



### A.6. Gambar Keseluruhan Alat



### **A.7. Biaya yang Dikeluarkan Pengunjung Tampil pada LCD**



### **A.8. Palang Wahana Terbuka**



-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN B

### B.1. Listing Program *Master* pada Loket

```
//Program Master

#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <Wire.h>

#define RST_PIN    9
#define SS_PIN    10
#define JUMLAH_ID  5
#define MODE      1 // 0 Cek Data Kartu - 1 Jalankan Program

//Inisialisasi RFID - LCD
MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal lcd(5,4,3,2,1,0);

//Deklarasi ID Pada Kartu
byte dataRFID[JUMLAH_ID][4] = {
    {0x84, 0xB3, 0x1F, 0x3B},
    {0x36, 0xBA, 0xEF, 0xA5},
    {0xB0, 0x64, 0x70, 0x85},
    {0x5A, 0x7E, 0xF6, 0xA5},
    {0xE0, 0x50, 0xC2, 0x1E}
};

//Deklarasi Nama Berdasarkan Kartu
char namaRFID[JUMLAH_ID][16] = {
    {"GUEST 1"},
    {"GUEST 2"},
    {"GUEST 3"},
    {"GUEST 4"},
    {"GUEST 5"}
};

//Deklarasi Global Variabel
unsigned int uangID[2]={0,0};
```

```

bool check = false;
bool foundID = false;
byte counterID, findID;

void rfidData(void);
void rfidCheck(void);
void rfidIdentification(void);
void sendToSlave(int, int);
void recFromSlave(int, int);

void setup() {
    SPI.begin();
    Wire.begin();
    mfr522.PCD_Init();
    lcd.begin(16,2);

    lcd.setCursor(0,0);
    lcd.print(" Bismillah ");
    lcd.setCursor(0,1);

    rfidCheck();
}

void loop() {
    MFRC522::MIFARE_Key key;
    for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
    lcd.print(" LOKET ");
    lcd.setCursor(0,1);
    lcd.print("Menunggu Kartu..");
    if ( ! mfr522.PICC_IsNewCardPresent() ) {
        return;
    }
    if ( ! mfr522.PICC_ReadCardSerial() ) {
        return;
    }
    lcd.clear();
    rfidCheck();
    lcd.print(" Card Detected ");
    lcd.setCursor(0,1);
}

```

```

    if(MODE){
        rfidIdentification();
    }else{ //mode 0
        rfidData();
    }

    delay(1000);
    lcd.clear();
}

//Fungsi yang digunakan untuk melihat data yang ada pada kartu
void rfidData(void){
    lcd.clear();
    for(byte i = 0; i < mfr522.uid.size; i++) {
        lcd.print(mfr522.uid.uidByte[i], HEX);
        lcd.print(' ');
    }
    delay(5000);
}

//Fungsi yang digunakan untuk melakukan pengecekan RC522
void rfidCheck(void){
    check =mfr522.PCD_PerformSelfTest();
    if(!check){
        lcd.print(" RFID not found ");
    }else{
        lcd.print(" RFID found ");
    }

    delay(500);
    lcd.clear();
    mfr522.PCD_Init();
}

//Fungsi yang digunakan untuk melakukan pengecekan apakah ID
ada atau tidak
void rfidIdentification(void){
    findID = 0; foundID = false;

```



```

while(!foundID && findID<JUMLAH_ID){ //Selama ID belum
ditemukan dan ID yang di cari kurang dari Total
    counterID=0;
    for(byte j = 0; j < mfrc522.uid.size; j++) {
        if(mfrc522.uid.uidByte[j] == dataRFID[findID][j])
            counterID++;
    }
    if(counterID>=4) foundID=true;
    else findID++;
}

if(foundID){ //Jika ID telah di temukan
    lcd.print("Teridentifikasi");
    delay(500);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Selamat Datang ");
    lcd.setCursor(0,1);
    lcd.print(namaRFID[findID]); //Menampilkan identitas pemilik
kartu
    delay(500);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Total ");
    sendToSlave(8,findID); //Mengirim sinyal ke slave dengan
alamat 8
    recFromSlave(8,findID); //Menerima jawaban dari slave
dengan alamat 8
    sendToSlave(9,findID); //Mengirim sinyal ke slave dengan
alamat 9
    recFromSlave(9,findID); //Menerima jawaban dari slave
dengan alamat 9
    lcd.setCursor(0,1);
    lcd.print("Rp. ");
    lcd.print(uangID[findID],DEC);
    lcd.print(" ribu");
    delay(500);
    uangID[findID]=0;
    //lcd.print(uangID[findID],DEC);

```

```

    }else{ //Jika ID tidak ditemukan
        lcd.print(" Tidak Terdaftar");
    }
}

//*****Fungsi Komunikasi I2C*****//
//Mengirimkan perintah ke slave
//slave -> Alamat Permainan
//ID -> ID Kartu

void sendToSlave(int slave, int num){
    Wire.beginTransmission(slave);
    Wire.write(num);
    Wire.endTransmission();
}

//Menerima tanggapan dari slave
void recFromSlave(int slave, int num){
    Wire.requestFrom(slave, 2);
    byte i = 0;
    unsigned char c[2];
    while(Wire.available()){
        c[i] = Wire.read();
        i++;
    }
    uangID[num] += ((c[1]<<8) | c[0]);
}

```

## B.2. Listing Program *slave* 1 pada Wahana 1

```

//Program Slave

#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include <Servo.h>

#define RST_PIN    9

```

```

#define SS_PIN      10
#define JUMLAH_ID  5
#define MODE        1 // 0 Cek Data Kartu - 1 Jalankan Program
#define BIAYA       45 //biaya wahana satuan dalam ribu

//Inisialisasi RFID - LCD
MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal lcd(5,4,3,2,1,0);
Servo myServo;

//Deklarasi ID Pada Kartu
byte dataRFID[JUMLAH_ID][4] ={
    {0x84, 0xB3, 0x1F, 0x3B},
    {0x36, 0xBA, 0xEF, 0xA5},
    {0xB0, 0x64, 0x70, 0x85},
    {0x5A, 0x7E, 0xF6, 0xA5},
    {0xE0, 0x50, 0xC2, 0x1E}
};

//Deklarasi Nama Berdasarkan Kartu
char namaRFID[JUMLAH_ID][16] ={
    {"GUEST 1"},
    {"GUEST 2"},
    {"GUEST 3"},
    {"GUEST 4"},
    {"GUEST 5"}
};

//Deklarasi Uang
int uang[JUMLAH_ID];

//Deklarasi Global Variabel
unsigned char data[2];
bool check = false;
bool foundID = false;
byte counterID, findID;
int ID;

//Deklarasi Fungsi
void rfidData(void);
void rfidCheck(void);

```

```

void rfidIdentification(void);

void setup() {
  SPI.begin();
  Wire.begin(8);
  Wire.onRequest(requestEvent);
  Wire.onReceive(receiveEvent);
  mfrc522.PCD_Init();
  myServo.attach(6); //Deklarasi pwm servo pada pin 6
  lcd.begin(16,2);

  lcd.setCursor(0,0);
  lcd.print(" Bismillah ");
  lcd.setCursor(0,1);
  lcd.print(" WAHANA 01 ");
  delay(1000);
  lcd.clear();

  rfidCheck();
  ID = 0;
}

void loop() {
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
  lcd.print(" WAHANA 01 ");
  lcd.setCursor(0,1);
  lcd.print("Menunggu Kartu..");
  if ( ! mfrc522.PICC_IsNewCardPresent() ) {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial() ) return;
  lcd.clear();

  rfidCheck();
  lcd.print(" Card Detected ");
  lcd.setCursor(0,1);
  if(MODE){
    rfidIdentification();
  }
}

```

```

    }else{
        rfidData();
    }

    delay(1000);
    lcd.clear();
}

```

```

//Fungsi yang digunakan untuk melihat data yang ada pada kartu
void rfidData(void){
    lcd.clear();
    for(byte i = 0; i < mfrc522.uid.size; i++) {
        lcd.print(mfrc522.uid.uidByte[i], HEX);
        lcd.print(' ');
    }
    delay(5000);
}

```

```

//Fungsi yang digunakan untuk melakukan pengecekan RC522
void rfidCheck(void){
    check =mfrc522.PCD_PerformSelfTest();
    if(!check){
        lcd.print(" RFID not found ");
        while(1);
    }else{
        lcd.print(" RFID found ");
    }
    /*while(!check){
        check =mfrc522.PCD_PerformSelfTest();
    }*/

    delay(500);
    lcd.clear();
    mfrc522.PCD_Init();
}

```

```

//Fungsi yang digunakan untuk melakukan pengecekan apakah ID
ada atau tidak
void rfidIdentification(void){

```

```

findID = 0; foundID = false;
while(!foundID && findID<JUMLAH_ID){
  counterID=0;
  for(byte j = 0; j < mfrc522.uid.size; j++) {
    if(mfrc522.uid.uidByte[j] == dataRFID[findID][j])
      counterID++;
  }
  if(counterID>=4) foundID=true;
  else          findID++;
}

if(foundID){
  lcd.print("Teridentifikasi");
  delay(500);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Selamat Datang ");
  lcd.setCursor(0,1);
  lcd.print(namaRFID[findID]);
  delay(500);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Total ");
  lcd.setCursor(0,1);
  lcd.print("Rp. ");
  uang[findID] += BIAAYA;

  lcd.print(uang[findID],DEC);
  lcd.print(" ribu");
  delay(100);
  //lcd.print(uangID[findID],DEC);
  gate();
}else{
  lcd.print(" Tidak Terdaftar");
}
}

//*****Fungsi Komunikasi I2C*****//
//Mengirimkan perintah ke slave

```

```

void receiveEvent(int howMany){
  ID = 0;
  ID = Wire.read();
}

void requestEvent(){
  data[0] = uang[ID] & 0xFF;
  data[1] = (uang[ID] >> 8) & 0xFF;
  Wire.write(data,2);
  uang[ID]=0;
}

//*****Fungsi Servo*****//
void gate(void){
  myServo.write(20); //tutup
  delay(2000); //bekerjapada 2ms
  myServo.write(90); //buka
  delay(2000);
  myServo.write(20); //tutup
  delay(2000);
}

```

### B.3. Listing Program *slave* 2 pada Wahana 2

```

//Program Slave

#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include <Servo.h>

#define RST_PIN    9
#define SS_PIN     10
#define JUMLAH_ID  5
#define MODE       1 // 0 Cek Data Kartu - 1 Jalankan Program
#define BIAYA      25 //biaya wahana satuan dalam ribu

//Inialisasi RFID - LCD

```

```

MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal lcd(5,4,3,2,1,0);
Servo myServo;

//Deklarasi ID Pada Kartu
byte dataRFID[JUMLAH_ID][4] = {
    {0x84, 0xB3, 0x1F, 0x3B},
    {0x36, 0xBA, 0xEF, 0xA5},
    {0xB0, 0x64, 0x70, 0x85},
    {0x5A, 0x7E, 0xF6, 0xA5},
    {0xE0, 0x50, 0xC2, 0x1E}
};
//Deklarasi Nama Berdasarkan Kartu
char namaRFID[JUMLAH_ID][16] = {
    {"GUEST 1"},
    {"GUEST 2"},
    {"GUEST 3"},
    {"GUEST 4"},
    {"GUEST 5"}
};
//Deklarasi Uang
int uang[JUMLAH_ID];

//Deklarasi Global Variabel
unsigned char data[2];
bool check = false;
bool foundID = false;
byte counterID, findID;
int ID;

//Deklarasi Fungsi
void rfidData(void);
void rfidCheck(void);
void rfidIdentification(void);

void setup() {
    SPI.begin();
    Wire.begin(9);
    Wire.onRequest(requestEvent);
}

```



```

Wire.onReceive(receiveEvent);
mfrc522.PCD_Init();
myServo.attach(6); //Deklarasi pwm servo pada pin 6
lcd.begin(16,2);

lcd.setCursor(0,0);
lcd.print(" Bismillah ");
lcd.setCursor(0,1);
lcd.print(" WAHANA 02 ");
delay(1000);
lcd.clear();

rfidCheck();
ID = 0;
}

void loop() {
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
  lcd.print(" WAHANA 02 ");
  lcd.setCursor(0,1);
  lcd.print("Menunggu Kartu..");
  if ( ! mfrc522.PICC_IsNewCardPresent() ) {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial() ) return;
  lcd.clear();

  rfidCheck();
  lcd.print(" Card Detected ");
  lcd.setCursor(0,1);
  if(MODE){
    rfidIdentification();
  }else{
    rfidData();
  }

  delay(1000);
  lcd.clear();
}

```

```

}

//Fungsi yang digunakan untuk melihat data yang ada pada kartu
void rfidData(void){
    lcd.clear();
    for(byte i = 0; i < mfrc522.uid.size; i++) {
        lcd.print(mfrc522.uid.uidByte[i], HEX);
        lcd.print(' ');
    }
    delay(5000);
}

```

```

//Fungsi yang digunakan untuk melakukan pengecekan RC522
void rfidCheck(void){
    check =mfrc522.PCD_PerformSelfTest();
    if(!check){
        lcd.print(" RFID not found ");
        while(1);
    }else{
        lcd.print(" RFID found ");
    }
    /*while(!check){
        check =mfrc522.PCD_PerformSelfTest();
    }*/

    delay(500);
    lcd.clear();
    mfrc522.PCD_Init();
}

```

//Fungsi yang digunakan untuk melakukan pengecekan apakah ID ada atau tidak

```

void rfidIdentification(void){
    findID = 0; foundID = false;
    while(!foundID && findID<JUMLAH_ID){
        counterID=0;
        for(byte j = 0; j < mfrc522.uid.size; j++) {
            if(mfrc522.uid.uidByte[j] == dataRFID[findID][j])
                counterID++;
        }
    }
}

```

```

    }
    if(counterID>=4) foundID=true;
    else          findID++;
}

if(foundID){
    lcd.print("Teridentifikasi");
    delay(500);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Selamat Datang ");
    lcd.setCursor(0,1);
    lcd.print(namaRFID[findID]);
    delay(500);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Total ");
    lcd.setCursor(0,1);
    lcd.print("Rp. ");
    uang[findID] += BIAYA;

    lcd.print(uang[findID],DEC);
    lcd.print(" ribu");
    delay(100);
    //lcd.print(uangID[findID],DEC);
    gate();
} else {
    lcd.print(" Tidak Terdaftar");
}
}

//*****Fungsi Komunikasi I2C*****//
//Mengirimkan perintah ke slave
void receiveEvent(int howMany){
    ID = 0;
    ID = Wire.read();
}

void requestEvent(){

```

```
data[0] = uang[ID] & 0xFF;
data[1] = (uang[ID] >> 8) & 0xFF;
Wire.write(data,2);
uang[ID]=0;
}
```

```
/*******Fungsi Servo*****//
void gate(void){
  myServo.write(20); //tutup
  delay(2000);
  myServo.write(90); //buka
  delay(2000);
  myServo.write(20); //tutup
  delay(2000);
}
```

-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN C

### C.1. Datasheet ATmega328

#### Features

- High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1K Bytes EEPROM
  - 512/1K/1K/2K Bytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA
  - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



8-bit AVR<sup>®</sup>  
Microcontroller  
with 4/8/16/32K  
Bytes In-System  
Programmable  
Flash

ATmega48A  
ATmega48PA  
ATmega88A  
ATmega88PA  
ATmega168A  
ATmega168PA  
ATmega328  
ATmega328P

Summary

Rev. 2071BS-AVR-0410





## 1.1 Pin Descriptions

### 1.1.1 VCC

Digital supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As Inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the Inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the Inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7...6 is used as TOSC2...1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in ["System Clock and Clock Options"](#) on page 26.

### 1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As Inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### 1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset Input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 29-12 on page 323](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ["Alternate Functions of Port C"](#) on page 86.

### 1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As Inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.



## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 89.

### 1.1.7 $AV_{CC}$

$AV_{CC}$  is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter. Note that PC6...4 use digital supply voltage,  $V_{CC}$ .

### 1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

### 1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

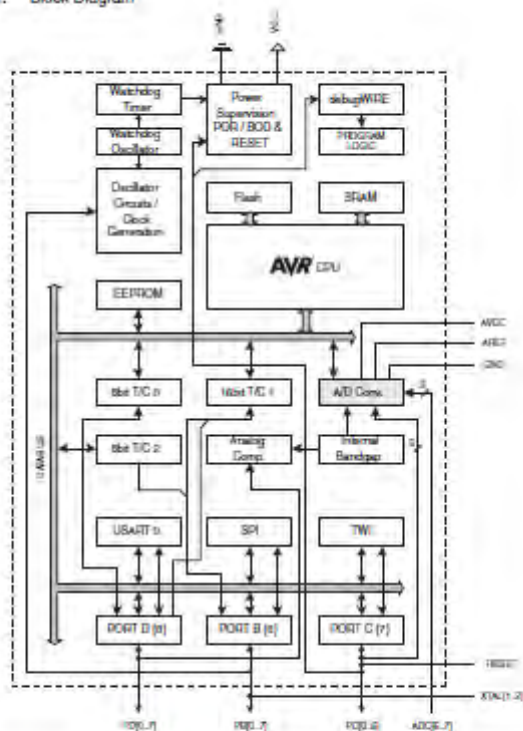
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

## 2. Overview

The ATmega48A/48PA/88A/88PA/168A/168PA/328/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48A/48PA/88A/88PA/168A/168PA/328/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

### 2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent

## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48A/48PA/88A/88PA/168A/168PA/328/328P provides the following features: 4K/8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, Internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with Internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48A/48PA/88A/88PA/168A/168PA/328/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48A/48PA/88A/88PA/168A/168PA/328/328P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation Kits.

### 2.2 Comparison Between Processors

The ATmega48A/48PA/88A/88PA/168A/168PA/328/328P differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48A	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88A	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168A	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector

## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

ATmega48A/48PA/88A/88PA/168A/168PA/328/328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega 48A/48PA there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

### 3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.







# ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1D (0x1D)	LIMSK	---	---	---	---	---	---	INT1	INT0	72
0x1C (0x1C)	EIFR	---	---	---	---	---	---	INT1	INT0	72
0x1B (0x1B)	PCRF	---	---	---	---	---	PCRF	PCRF	PCRF	
0x1A (0x1A)	Reserved	---	---	---	---	---	---	---	---	
0x19 (0x19)	Reserved	---	---	---	---	---	---	---	---	
0x18 (0x18)	Reserved	---	---	---	---	---	---	---	---	
0x17 (0x17)	IFR2	---	---	---	---	---	OCF2B	OCF2A	TOVF	84
0x16 (0x16)	IFR1	---	---	ICF1	---	---	OCF1B	OCF1A	TOVF	84
0x15 (0x15)	IFR0	---	---	---	---	---	OCF0B	OCF0A	TOVF	
0x14 (0x14)	Reserved	---	---	---	---	---	---	---	---	
0x13 (0x13)	Reserved	---	---	---	---	---	---	---	---	
0x12 (0x12)	Reserved	---	---	---	---	---	---	---	---	
0x11 (0x11)	Reserved	---	---	---	---	---	---	---	---	
0x10 (0x10)	Reserved	---	---	---	---	---	---	---	---	
0x0F (0x0F)	Reserved	---	---	---	---	---	---	---	---	
0x0E (0x0E)	Reserved	---	---	---	---	---	---	---	---	
0x0D (0x0D)	Reserved	---	---	---	---	---	---	---	---	
0x0C (0x0C)	Reserved	---	---	---	---	---	---	---	---	
0x0B (0x0B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	84
0x0A (0x0A)	DDRD	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	84
0x09 (0x09)	PNP0	PNP07	PNP06	PNP05	PNP04	PNP03	PNP02	PNP01	PNP00	84
0x08 (0x08)	PORTC	---	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	82
0x07 (0x07)	DDRC	---	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	82
0x06 (0x06)	PNP1	---	PNP16	PNP15	PNP14	PNP13	PNP12	PNP11	PNP10	82
0x05 (0x05)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	82
0x04 (0x04)	DDRB	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	82
0x03 (0x03)	PNP2	PNP27	PNP26	PNP25	PNP24	PNP23	PNP22	PNP21	PNP20	82
0x02 (0x02)	Reserved	---	---	---	---	---	---	---	---	
0x01 (0x01)	Reserved	---	---	---	---	---	---	---	---	
0x00 (0x00)	Reserved	---	---	---	---	---	---	---	---	

- Note:
- For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
  - I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
  - Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
  - When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x2F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48A/48PA/88A/88PA/168A/168PA/328/328P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
  - Only valid for ATmega88A/88PA/168A/168PA/328/328P.
  - BODS and BODSE only available for picoPower devices ATmega48PA/88PA/168PA/328P.





5. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Cycles
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z, C, N, V, H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z, C, N, V, H	1
ADDFW	Rd, K	Add Immediate to Word	Rd ← Rd + RdHigh + RdLow + K	Z, C, N, V, S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z, C, N, V, H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z, C, N, V, H	1
SUBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z, C, N, V, H	1
SUBCW	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z, C, N, V, H	1
SUBFW	Rd, K	Subtract Immediate from Word	Rd ← Rd - RdHigh - RdLow - K	Z, C, N, V, S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd & Rr	Z, N, V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd & K	Z, N, V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd   Rr	Z, N, V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd   K	Z, N, V	1
ECOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z, N, V	1
ECORI	Rd, K	Exclusive OR Register and Constant	Rd ← Rd ⊕ K	Z, N, V	1
COM	Rd	One's Complement	Rd ← ~Rd	Z, C, N, V	1
NEG	Rd	Two's Complement	Rd ← ~Rd + 1	Z, C, N, V, H	1
SEI	Rd, K	Set Bit(s) in Register	Rd ← Rd   K	Z, N, V	1
CLI	Rd, K	Clear Bit(s) in Register	Rd ← Rd & ~K	Z, N, V	1
INC	Rd	Increment	Rd ← Rd + 1	Z, N, V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z, N, V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd & Rd	Z, N, V	1
CLR	Rd	Clear Register	Rd ← Rd & 0	Z, N, V	1
SETB	Rd	Set Register	Rd ← Rd   1	None	1
MUL	Rd, Rr	Multiply Unsigned	R16 ← Rd × Rr	Z, C	2
MULS	Rd, Rr	Multiply Signed	R16 ← Rd × Rr	Z, C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R16 ← Rd × Rr	Z, C	2
FIMUL	Rd, Rr	Fractional Multiply Unsigned	R16 ← (Rd × Rr) × 1	Z, C	2
FIMULS	Rd, Rr	Fractional Multiply Signed	R16 ← (Rd × Rr) × 1	Z, C	2
FIMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R16 ← (Rd × Rr) × 1	Z, C	2
<b>BRANCH INSTRUCTIONS</b>					
JMP	k	Relative Jump	PC ← PC + k + 1	None	2
LDI	k	Indirect Jump to (Z)	PC ← Z	None	2
RETI	k	Return Jump	PC ← k	None	2
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	2
CALL	k	Indirect Call to (Z)	PC ← Z	None	4
CALLR	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETR		Interrupt Return	PC ← STACK	I	4
CPSE	Rd/Rr	Compare, Skip if Equal	#(Rd - Rr)   PC ← PC + 2 or 3	None	10/2
CP	Rd/Rr	Compare	Rd - Rr	Z, N, V, C, H	1
CPC	Rd/Rr	Compare with Carry	Rd - Rr - C	Z, N, V, C, H	1
CPY	Rd, K	Compare Register with Immediate	Rd - K	Z, N, V, C, H	1
SBSC	Rr, n	Skip if Bit in Register Cleared	#(Rd[n] & 0)   PC ← PC + 2 or 3	None	10/2
SBSE	Rr, n	Skip if Bit in Register Set	#(Rd[n] & 1)   PC ← PC + 2 or 3	None	10/2
SBSC	P, b	Skip if Bit in I/O Register Cleared	#(SPDR[b] & 0)   PC ← PC + 2 or 3	None	10/2
SBSE	P, b	Skip if Bit in I/O Register Set	#(SPDR[b] & 1)   PC ← PC + 2 or 3	None	10/2
SBSC	a, k	Search if Status Flag Cleared	#(SREG[a] & 0)   Branch PC ← PC + k + 1	None	1/2
SBSC	a, k	Search if Status Flag Cleared	#(SREG[a] & 0)   Branch PC ← PC + k + 1	None	1/2
SBSC	k	Search if Equal	#(Z = 1)   Branch PC ← PC + k + 1	None	1/2
SBNE	k	Search if Not Equal	#(Z = 0)   Branch PC ← PC + k + 1	None	1/2
SBCS	k	Search if Carry Set	#(C = 1)   Branch PC ← PC + k + 1	None	1/2
SBCC	k	Search if Carry Cleared	#(C = 0)   Branch PC ← PC + k + 1	None	1/2
SBHS	k	Search if Same or Higher	#(C = 0)   Branch PC ← PC + k + 1	None	1/2
SBLO	k	Search if Lower	#(C = 1)   Branch PC ← PC + k + 1	None	1/2
SBMI	k	Search if Minus	#(N = 1)   Branch PC ← PC + k + 1	None	1/2
SBPL	k	Search if Plus	#(N = 0)   Branch PC ← PC + k + 1	None	1/2
SBGE	k	Search if Greater or Equal, Signed	#(N & V = 0)   Branch PC ← PC + k + 1	None	1/2
SBGT	k	Search if Less Than Zero, Signed	#(N & V = 1)   Branch PC ← PC + k + 1	None	1/2
SBHS	k	Search if Half Carry Flag Set	#(H = 1)   Branch PC ← PC + k + 1	None	1/2
SBHC	k	Search if Half Carry Flag Cleared	#(H = 0)   Branch PC ← PC + k + 1	None	1/2
SBIF	k	Search if I Flag Set	#(I = 1)   Branch PC ← PC + k + 1	None	1/2
SBIF	k	Search if I Flag Cleared	#(I = 0)   Branch PC ← PC + k + 1	None	1/2
SBIF	k	Search if Overflow Flag Set	#(V = 1)   Branch PC ← PC + k + 1	None	1/2
SBIF	k	Search if Overflow Flag is Cleared	#(V = 0)   Branch PC ← PC + k + 1	None	1/2



# ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

Mnemonics	Operands	Description	Operation	Flags	#Clocks
IREI	k	Search if Interrupt Enabled	$I \leftarrow 0$ if $PC = PC + k + 1$	None	10
IRRI	k	Search if Interrupt Disabled	$I \leftarrow 0$ if $\text{then } PC = PC + k + 1$	None	10
<b>BIT AND BIT-TEST INSTRUCTIONS</b>					
SBIC	P.b	Set Bit in I/O Register	$IOP(b) = 1$	None	2
SBSI	P.b	Clear Bit in I/O Register	$IOP(b) = 0$	None	2
LDI	Rd	Logical Shift Left	$Rd(b) \leftarrow Rd(b), Rd(b) \leftarrow 0$	Z,C,N,V	1
LRI	Rd	Logical Shift Right	$Rd(b) \leftarrow Rd(b), Rd(7) \leftarrow 0$	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	$Rd(b) \leftarrow C, Rd(b) \leftarrow Rd(b)$	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	$Rd(7) \leftarrow C, Rd(7) \leftarrow Rd(7), C \leftarrow Rd(7)$	Z,C,N,V	1
ASRF	Rd	Arithmetic Shift Right	$Rd(b) \leftarrow Rd(b), b \leftarrow 0$	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	$Rd(3:0) \leftrightarrow Rd(7:4), Rd(7:4) \leftrightarrow Rd(3:0)$	None	1
SREI	a	Flag Clear	$SREG(a) = 0$	SREG(a)	1
SSEI	a	Flag Clear	$SREG(a) = 1$	SREG(a)	1
SBIC	Rd.b	SB Test from Register to I	$T \leftarrow Rd(b)$	T	1
SRLD	Rd.b	SB Test from I to Register	$Rd(b) = T$	None	1
SFIC		Set Carry	$C = 1$	C	1
CLD		Clear Carry	$C = 0$	C	1
SENB		Set Negative Flag	$N = 1$	N	1
CLN		Clear Negative Flag	$N = 0$	N	1
SEZ		Set Zero Flag	$Z = 1$	Z	1
CLZ		Clear Zero Flag	$Z = 0$	Z	1
SEI		Global Interrupt Enable	$I = 1$	I	1
CLI		Global Interrupt Disable	$I = 0$	I	1
SEIS		Set Signed Test Flag	$S = 1$	S	1
CLS		Clear Signed Test Flag	$S = 0$	S	1
SEV		Set Two Complement Overflow	$V = 1$	V	1
CLV		Clear Two Complement Overflow	$V = 0$	V	1
SET		Set T in SREG	$T = 1$	T	1
CLT		Clear T in SREG	$T = 0$	T	1
SEH		Set Half Carry Flag in SREG	$H = 1$	H	1
CLH		Clear Half Carry Flag in SREG	$H = 0$	H	1
<b>DATA TRANSFER INSTRUCTIONS</b>					
MOV	Rd, Rr	Move between Registers	$Rd \leftarrow Rr$	None	1
MOVW	Rd, Rr	Copy Register Word	$Rd \leftarrow Rr, Rr \leftarrow Rr$	None	1
LDI	Rd, k	Load Immediate	$Rd \leftarrow k$	None	1
LDD	Rd, X	Load Indirect	$Rd \leftarrow [X]$	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	$Rd \leftarrow [X], X \leftarrow X + 1$	None	2
LD	Rd, X-	Load Indirect and Pre-Dec.	$X \leftarrow X - 1, Rd \leftarrow [X]$	None	2
LD	Rd, Y	Load Indirect	$Rd \leftarrow [Y]$	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	$Rd \leftarrow [Y], Y \leftarrow Y + 1$	None	2
LD	Rd, Y-	Load Indirect and Pre-Dec.	$Y \leftarrow Y - 1, Rd \leftarrow [Y]$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow [Z + q]$	None	2
LD	Rd, Z	Load Indirect	$Rd \leftarrow [Z]$	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	$Rd \leftarrow [Z], Z \leftarrow Z + 1$	None	2
LD	Rd, Z-	Load Indirect and Pre-Dec.	$Z \leftarrow Z - 1, Rd \leftarrow [Z]$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow [Z + q]$	None	2
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow [k]$	None	2
ST	X, Rr	Store Indirect	$[X] \leftarrow Rr$	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$[X] \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	X-, Rr	Store Indirect and Pre-Dec.	$X \leftarrow X - 1, [X] \leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$[Y] \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$[Y] \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	Y-, Rr	Store Indirect and Pre-Dec.	$Y \leftarrow Y - 1, [Y] \leftarrow Rr$	None	2
STD	Y+q, Rr	Store Indirect with Displacement	$[Y + q] \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$[Z] \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	$[Z] \leftarrow Rr, Z \leftarrow Z + 1$	None	2
ST	Z-, Rr	Store Indirect and Pre-Dec.	$Z \leftarrow Z - 1, [Z] \leftarrow Rr$	None	2
STD	Z+q, Rr	Store Indirect with Displacement	$[Z + q] \leftarrow Rr$	None	2
STS	k, Rr	Store Direct to SRAM	$[k] \leftarrow Rr$	None	2
LPM		Load Program Memory	$Rd \leftarrow [Z]$	None	2
LPM	Rd, Z	Load Program Memory	$Rd \leftarrow [Z]$	None	2
LPM	Rd, Z+	Load Program Memory and Post-Inc.	$Rd \leftarrow [Z], Z \leftarrow Z + 1$	None	2
SFM		Store Program Memory	$[Z] \leftarrow R150$	None	2
IN	Rd, P	In Port	$Rd \leftarrow P$	None	1
OUT	P, Rr	Out Port	$P \leftarrow Rr$	None	1
PUSH	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2



## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

Mnemonics	Operands	Description	Operation	Flags	#Cycles
POP	Rd	Copy Data to Register	Rd ← STACK	None	2
<b>MCU CONTROL INSTRUCTIONS</b>					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descn for Sleep function)	None	1
WDN		Watchdog Reset	(see specific descn for WDT/Timer)	None	1
BRIEAK		Break	For On-chip Debug Only	None	1/2

Note: 1. These instructions are only available in ATmega168PA and ATmega328P.

# ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

## 6. Ordering Information

### 6.1 ATmega48A

Speed (MHz)	Power Supply	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(3)</sup>	1.8 - 5.5	ATmega48A-AU ATmega48A-AUH <sup>(4)</sup> ATmega48A-MMH <sup>(4)</sup> ATmega48A-MMHR <sup>(4)(5)</sup> ATmega48A-MU ATmega48A-MUR <sup>(4)</sup> ATmega48A-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See "Speed Grades" on page 321.
  4. NiPdAu Lead Finish.
  5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)

# ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

## 6.2 ATmega48PA

Speed (MHz)	Power Supply	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(3)</sup>	1.8 - 5.5	ATmega48PA-AJ ATmega48PA-AJR <sup>(4)</sup> ATmega48PA-MMH <sup>(4)</sup> ATmega48PA-MMHR <sup>(4)(5)</sup> ATmega48PA-MU ATmega48PA-MUR <sup>(3)</sup> ATmega48PA-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in water form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See "Speed Grades" on page 321.
  4. NiPdAu Lead Finish.
  5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)

## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

### 6.3 ATmega88A

Speed (MHz)	Power Supply	Ordering Code <sup>(1)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(2)</sup>	1.8 - 5.5	ATmega88A-AU ATmega88A-AUH <sup>(3)</sup> ATmega88A-MM <sup>(4)</sup> ATmega88A-MM-IR <sup>(4)(5)</sup> ATmega88A-MU ATmega88A-MUR <sup>(5)</sup> ATmega88A-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See "Speed Grades" on page 321.
  4. NiPdAu Lead Finish.
  5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual In-Line Package (PDIP)

## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

### 6.4 ATmega88PA

Speed (MHz)	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(2)</sup>	1.8 - 5.5	ATmega88PA-AJ ATmega88PA-AJF <sup>(3)</sup> ATmega88PA-MMH <sup>(4)</sup> ATmega88PA-MMHF <sup>(4)(5)</sup> ATmega88PA-MU ATmega88PA-MUF <sup>(2)</sup> ATmega88PA-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See "Speed Grades" on page 321.
  4. NiPdAu Lead Finish.
  5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)

## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

### 6.5 ATmega168A

Speed (MHz) <sup>(1)</sup>	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20	1.8 - 5.5	ATmega168A-AU ATmega168A-AUH <sup>(3)</sup> ATmega168A-MMH <sup>(4)</sup> ATmega168A-MMHR <sup>(4)(5)</sup> ATmega168A-MU ATmega168A-MUR <sup>(3)</sup> ATmega168A-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
- This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  - Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  - See "Speed Grades" on page 321
  - NiPdAu Lead Finish.
  - Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)





## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

### 6.6 ATmega168PA

Speed (MHz) <sup>(1)</sup>	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20	1.8 - 5.5	ATmega168PA-AU ATmega168PA-AUH <sup>(3)</sup> ATmega168PA-MMH <sup>(4)</sup> ATmega168PA-MMHR <sup>(4)(5)</sup> ATmega168PA-MJ ATmega168PA-MUR <sup>(3)</sup> ATmega168PA-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

Note: 1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. See "Speed Grades" on page 321.
4. NiPdAu Lead Finish.
5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)

## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

### 6.7 ATmega328

Speed (MHz)	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(3)</sup>	1.8 - 5.5	ATmega328-AU ATmega328-AUR <sup>(4)</sup> ATmega328-MU ATmega328-MUR <sup>(4)</sup> ATmega328-PU	32A 32A 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See [Figure 28-1](#) on [page 321](#).
  4. Tape & Reel

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28P3	28-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

## ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

### 6.8 ATmega328P

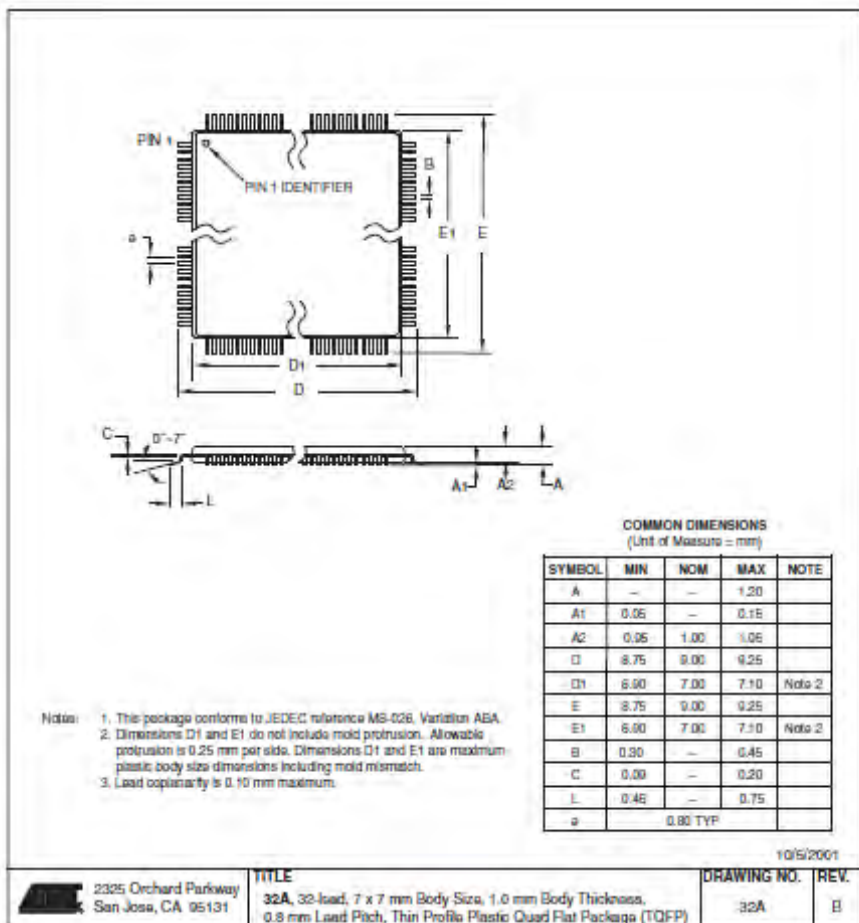
Speed (MHz)	Power Supply	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(3)</sup>	1.8 - 5.5	ATmega328P-AJ ATmega328P-AJR <sup>(4)</sup> ATmega328P-MJ ATmega328P-MJR <sup>(4)</sup> ATmega328P-PU	32A 32A 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See Figure 28-1 on page 321.
  4. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28P3	28-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

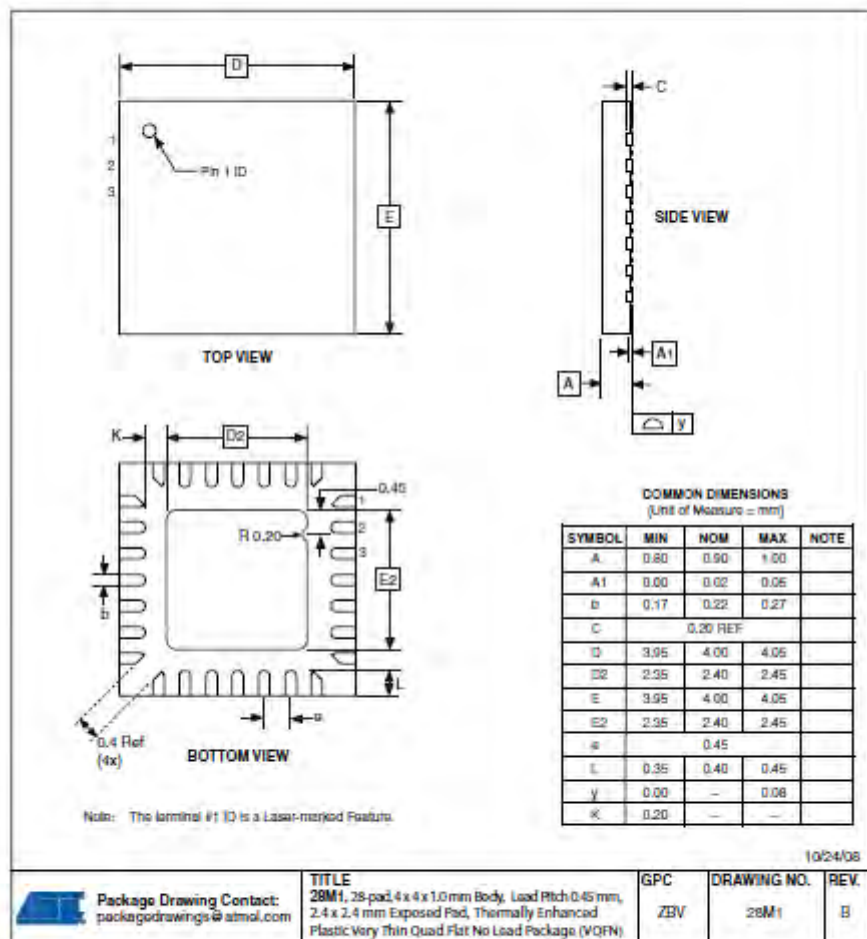
## 7. Packaging Information

### 7.1 32A



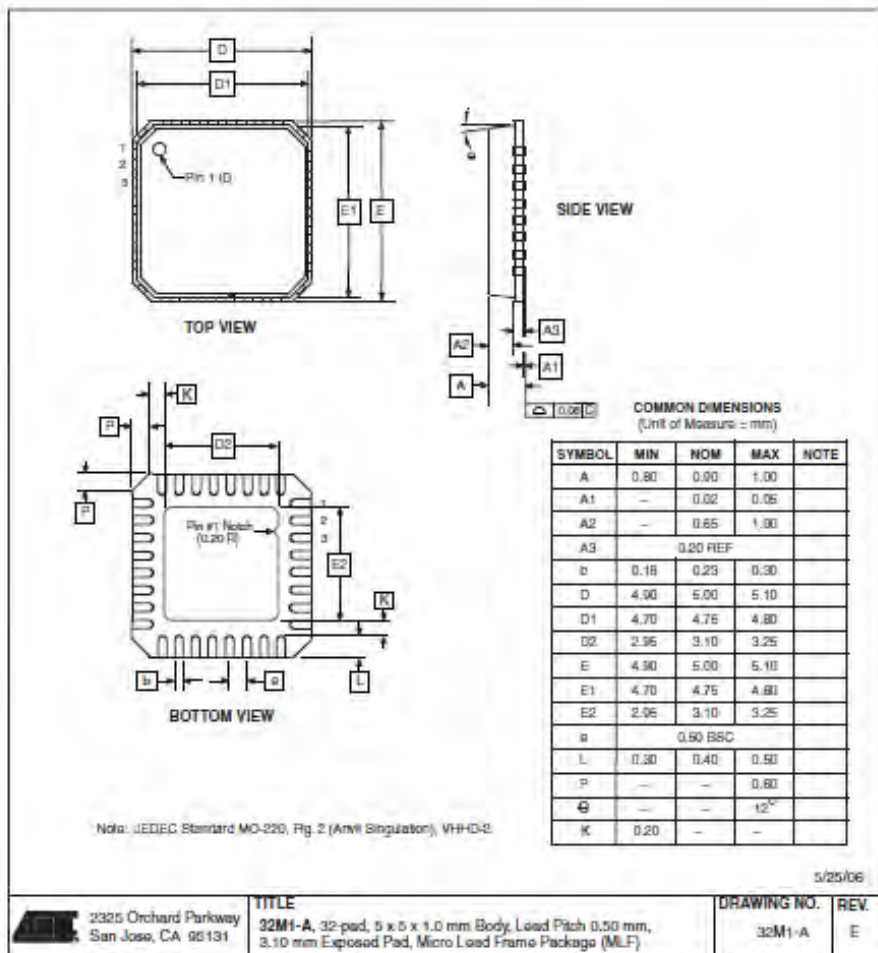
# ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

## 7.2 28M1



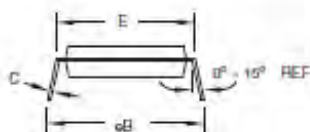
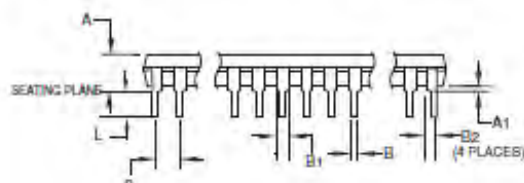
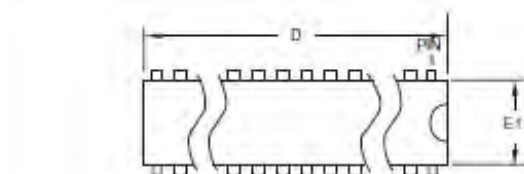
# ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

## 7.3 32M1-A



# ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

## 7.4 28P3



COMMON DIMENSIONS  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	—	—	4.5724	
A1	0.508	—	—	
D	34.544	—	34.798	Note 1
E	7.620	—	8.255	
E1	7.112	—	7.499	Note 1
B	0.381	—	0.533	
B1	1.143	—	1.327	
B2	0.762	—	1.143	
L	3.175	—	3.429	
C	0.203	—	0.352	
e	—	—	16.160	
e	2.540 TYP			

Note: 1. Dimensions D and E1 do not include mold flash or protrusion. Mold flash or protrusion shall not exceed 0.25 mm (0.010").

09/28/01



2325 Orchard Parkway  
San Jose, CA 95131

TITLE

28P3, 28-lead (0.300"/7.62 mm Wide) Plastic Dual In-line Package (PDIP)

DRAWING NO.

28P3

REV.

B



## 8. Errata

### 8.1 Errata ATmega48A

The revision letter in this section refers to the revision of the ATmega48A device.

#### 8.1.1 Rev. D

- **Analog MUX can be turned off when setting ACME bit**
  1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**  
Clear the MUX3 bit before setting the ACME bit.

### 8.2 Errata ATmega48PA

The revision letter in this section refers to the revision of the ATmega48PA device.

#### 8.2.1 Rev. D

- **Analog MUX can be turned off when setting ACME bit**
  1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**  
Clear the MUX3 bit before setting the ACME bit.

### 8.3 Errata ATmega88A

The revision letter in this section refers to the revision of the ATmega88A device.

#### 8.3.1 Rev. F

- **Analog MUX can be turned off when setting ACME bit**
  1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**  
Clear the MUX3 bit before setting the ACME bit.



## 8.4 Errata ATmega88PA

The revision letter in this section refers to the revision of the ATmega88PA device.

### 8.4.1 Rev. F

• Analog MUX can be turned off when setting ACME bit

1. Analog MUX can be turned off when setting ACME bit

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]~1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

## 8.5 Errata ATmega168A

The revision letter in this section refers to the revision of the ATmega168A device.

### 8.5.1 Rev. E

• Analog MUX can be turned off when setting ACME bit

1. Analog MUX can be turned off when setting ACME bit

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]~1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

## 8.6 Errata ATmega168PA

The revision letter in this section refers to the revision of the ATmega168PA device.

### 8.6.1 Rev. E

• Analog MUX can be turned off when setting ACME bit

1. Analog MUX can be turned off when setting ACME bit

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]~1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.



## 8.7 Errata ATmega328

The revision letter in this section refers to the revision of the ATmega328 device.

### 8.7.1 Rev D

- Analog MUX can be turned off when setting ACME bit
- 1. **Analog MUX can be turned off when setting ACME bit**  
If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.  
**Problem Fix/Workaround**  
Clear the MUX3 bit before setting the ACME bit.

### 8.7.2 Rev C

Not sampled.

### 8.7.3 Rev B

- Analog MUX can be turned off when setting ACME bit
- Unstable 32 kHz Oscillator
- 1. **Unstable 32 kHz Oscillator**  
If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.  
**Problem Fix/Workaround**  
Clear the MUX3 bit before setting the ACME bit.
- 2. **Unstable 32 kHz Oscillator**  
The 32 kHz oscillator does not work as system clock. The 32 kHz oscillator used as asynchronous timer is inaccurate.  
**Problem Fix/ Workaround**  
None.

### 8.7.4 Rev A

- Analog MUX can be turned off when setting ACME bit
- Unstable 32 kHz Oscillator
- 1. **Unstable 32 kHz Oscillator**  
If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.  
**Problem Fix/Workaround**  
Clear the MUX3 bit before setting the ACME bit.
- 2. **Unstable 32 kHz Oscillator**  
The 32 kHz oscillator does not work as system clock. The 32 kHz oscillator used as asynchronous timer is inaccurate.  
**Problem Fix/ Workaround**  
None.



## 8.8 Errata ATmega328P

The revision letter in this section refers to the revision of the ATmega328P device.

### 8.8.1 Rev D

- **Analog MUX can be turned off when setting ACME bit**
- 1. **Analog MUX can be turned off when setting ACME bit**  
If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.  
**Problem Fix/Workaround**  
Clear the MUX3 bit before setting the ACME bit.

### 8.8.2 Rev C

Not sampled.

### 8.8.3 Rev B

- **Analog MUX can be turned off when setting ACME bit**
- **Unstable 32 kHz Oscillator**
- 1. **Unstable 32 kHz Oscillator**  
If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.  
**Problem Fix/Workaround**  
Clear the MUX3 bit before setting the ACME bit.
- 2. **Unstable 32 kHz Oscillator**  
The 32 kHz oscillator does not work as system clock. The 32 kHz oscillator used as asynchronous timer is inaccurate.  
**Problem Fix/ Workaround**  
None.

### 8.8.4 Rev A

- **Unstable 32 kHz Oscillator**
- 1. **Unstable 32 kHz Oscillator**  
The 32 kHz oscillator does not work as system clock. The 32 kHz oscillator used as asynchronous timer is inaccurate.  
**Problem Fix/ Workaround**  
None.

## 9. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

### 9.1 Rev. 8271B-04/10

1. Updated Table 8-6 with correct value for timer oscillator at xtal2/xtal2
2. Corrected use of SBIS instructions in assembly code examples.
3. Corrected BOD and BODSE bits to RW in Section 9.11.2 on page 45, Section 11.5 on page 69 and Section 13.4 on page 93
4. Figures for bandgap characterization added, Figure 29-34 on page 349, Figure 29-81 on page 374, Figure 29-128 on page 398, Figure 29-175 on page 424, Figure 29-222 on page 440, Figure 29-269 on page 474, Figure 29-316 on page 499 and Figure 29-363 on page 523.
5. Updated "Packaging Information" on page 546 by replacing 28M1 with a correct corresponding package.

### 9.2 Rev. 8271A-12/09

1. New datasheet 8271 with merged information for ATmega48PA, ATmega88PA, ATmega168PA and ATmega48A, ATmega88A and ATmega168A. Also included information on ATmega328 and ATmega328P
2. Changes done:
  - New devices added: ATmega48A/ATmega88A/ATmega168A and ATmega328
  - Updated Feature Description
  - Updated Table 2-1 on page 6
  - Added note for BOD Disable on page 40.
  - Added note on BOD and BODSE in "MCUCR – MCU Control Register" on page 93 and "Register Description" on page 294
  - Added limitation information for the application "Boot Loader Support – Read-While-Write Self-Programming" on page 279
  - Added limitation information for "Program And Data Memory Lock Bits" on page 296
  - Added specified DC characteristics per processor
  - Added typical characteristics per processor
  - Removed exception information in "Address Match Unit" on page 225.



## Headquarters

---

**Atmel Corporation**  
2325 Orchard Parkway  
San Jose, CA 95131  
USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 487-2600

## International

---

**Atmel Asia**  
Unit 1-5 & 16, 19/F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
Hong Kong  
Tel: (852) 2245-6100  
Fax: (852) 2722-1380

**Atmel Europe**  
Le Krebs  
8, Rue Jean-Pierre Timbaud  
BP 300  
78054 Saint-Quentin-en-  
Yvelines Cedex  
France  
Tel: (33) 1-30-60-70-00  
Fax: (33) 1-30-60-71-11

**Atmel Japan**  
6F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0093  
Japan  
Tel: (81) 3-3523-3511  
Fax: (81) 3-3523-7581

## Product Contact

---

**Web Site**  
[www.atmel.com](http://www.atmel.com)

**Technical Support**  
[avr@atmel.com](mailto:avr@atmel.com)

**Sales Contact**  
[www.atmel.com/contacts](http://www.atmel.com/contacts)

**Literature Requests**  
[www.atmel.com/litreq](http://www.atmel.com/litreq)

---

**Disclaimer:** The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2010 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, AVR®, AVR® logo and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

82718S-AVR-04/10

## C.2. Datasheet Motor Servo

### S3003 FUTABA SERVO



#### ...S3003 FUTABA SERVO...

##### Detailed Specifications

Control System:	+Pulse Width Control 1520usec: Neutral	Current Drain (4.8V):	7.2mA/Idle
Required Pulse:	3-5 Volt Peak to Peak Square Wave	Current Drain (6.0V):	8mA/Idle
Operating Voltage:	4.8-6.0 Volts	Direction:	Counter Clockwise/Pulse Traveling 1520- 1900usec
Operating Temperature Range:	-20 to +60 Degree C	Motor Type:	3 Pole Ferrite
Operating Speed (4.8V):	0.23sec/60 degrees at no load	Potentiometer Drive:	Indirect Drive
Operating Speed (6.0V):	0.19sec/60 degrees at no load	Bearing Type:	Plastic Bearing
Stall Torque (4.8V):	44 oz/in. (3.2kg.cm)	Gear Type:	All Nylon Gears
Stall Torque (6.0V):	56.8 oz/in. (4.1kg.cm)	Connector Wire Length:	12"
Operating Angle:	45 Deg. one side pulse traveling 400usec	Dimensions:	1.6" x 0.8" x 1.4" (41 x 20 x 36mm)
360 Modifiable:	Yes	Weight:	1.3oz. (37.2g)

-----Halaman ini sengaja dikosongkan-----

## **BAB V**

### **PENUTUP**

Dari perancangan dan pengujian alat, dapat ditarik kesimpulan dan saran mengenai Tugas Akhir berikut:

#### **5.1 Kesimpulan**

1. RFID *reader* RC522 yang terdapat pada alat dapat membaca *tag* RFID pada jarak kurang dari 4cm, dan akan mendeteksi secara maksimal ketika pengecekan *tag* dilakukan pada posisi sejajar terhadap *reader*.
2. Pembacaan kartu *tag* RFID dapat mempengaruhi tegangan *input* pada RFID *reader*, di mana pada tiap-tiap wahana dan loket pembayaran memiliki tegangan *input* yang berbeda-beda pada RFID *reader*.
3. RFID *reader* ketika tidak didekatkan pada RFID *tag* maka tegangan *input* yaitu 3,68 V, sedangkan pada saat keadaan RFID *tag* didekatkan dengan RFID *reader* maka tegangan *input* bertambah maksimal menjadi 3,84 V.
4. *Prototype* sistem pembayaran dengan metode pasca bayar menggunakan RFID dapat menjalankan mekanisme pintu gerbang dengan sudut saat pintu gerbang terbuka yaitu 90° dan sudut saat pintu gerbang tertutup yaitu 0°.

#### **5.2 Saran**

1. Pengembangan dari Tugas Akhir ini agar lebih sempurna dapat dilakukan dengan cara yaitu menambahkan laporan pada setiap transaksi untuk meningkatkan kenyamanan dan kebutuhan pengunjung.
2. Pada komunikasi tiap-tiap wahana dapat juga diganti dengan menggunakan metode *wireless*.



-----Halaman ini sengaja dikosongkan-----

## DAFTAR PUSTAKA

- [1] Dyas Ayu Restina dan Subhan Hadfin Permana Putra, “Rancang Bangun Sistem Pembayaran Pada Wahana Permainan Dengan RFID”, *Tugas Akhir*, Program Studi D3 Teknik Elektro FTI-ITS, Surabaya, 2012
- [2] Korneel, *Getting RFID RC-522 to Work!*, <URL: <https://community.particle.io/t/getting-the-rfid-rc522-to-work-solved/3571>>, 14 Juni 2016.
- [3] Unun Nindya Ardani dan Thorikul Huda, “Perancangan dan Pembuatan Sistem Pembayaran pada Pengisian Bensin Otomatis menggunakan Sensor RFID”, *Tugas Akhir*, Program Studi D3 Teknik Elektro FTI-ITS, Surabaya, 2009
- [4] Muhammad Yanuar Effendi dan Ahmad Khanafi, “Penggunaan Sensor RFID untuk Tiket Prabayar Jalan Bebas Hambatan”, *Tugas Akhir*, Program Studi D3 Teknik Elektro FTI-ITS, Surabaya, 2010
- [5] Wilfried Voss. *A Comprehensible Guide to Servo Motor Sizing*. Greenfield: *Copperhill Technologies Corporation*. 2007
- [6] Frances Reed, *How Do Servo Motors Work*, <URL: <http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>>, 14 Juni 2016.
- [7] Trikueni Dermanto, *Pengertian dan Prinsip Kerja Motor Servo*, <URL:<http://trikueni-desain-sistem.blogspot.co.id/2014/03/Pengertian-Motor-Servo.html>>, 14 Juni 2016.
- [8] Steven F. Barret. *Arduino Microcontroller Processing for Everyone! Third Edition*. Wyoming: Morgan & Claypool Publishers. 2013
- [9] ....., *ATMEGA328P-PU Atmel 8 Bit 32K AVR Microcontroller*, NKCElektronik <URL: <http://www.prostotack.com/micro-controllers/atmega328p-pu-atmel-8-bit-32k-avr-microcontroller>>, 14 Juni 2016.
- [10] Eduardo Pilo. *Power Supply, Energy Management and Canetary Problems*. Southampton: WIT Press. 2010
- [11] Widodo Budiharto & Togu Jefri. *12 Proyek Sistem Akuisisi Data*. Jakarta: PT Elex Media Komputindo. 2007
- [12] Purnomo Sejati, *Sinyal ACK dan NACK*, <URL: <https://purnomosejati.com/2011/08/25/mengenal-komunikasi2cinter-integrated-circuit/>>, 14 Juni 2016.

- [13] Purnomo Sejati, *Transfer Bit Pada I2C Bus*, <URL: <https://purnomosejati.com/2011/08/25/mengenal-komunikasi-i2cinter-integrated-circuit/>>, 14 Juni 2016.
- [14] Purnomo Sejati, *Mengenal Komunikasi I2C(Inter Integrated Circuit)*, <URL: <https://purnomosejati.com/2011/08/25/mengenal-komunikasi-i2cinter-integrated-circuit/>>, 14 Juni 2016.
- [15] Priyadi Utomo. *Fisika Interaktif*. Jakarta: Azka Press. 2007
- [16] ....., *LM2596 DC-DC Step-Down Module*, Electrafreak <URL: <http://www.electfreaks.com/store/lm2596-dcdc-stepdown-module-bkdc010-p-292.html>>, 14 Juni 2016.
- [17] Addison Kimbali, *A little "P" makes a lot of difference*, <URL: <http://thewindycitylab.com/little-p-makes-lot-difference/>>, 14 Juni 2016.
- [18] Brendan, *RFID and Inventory Control*, <URL: <http://adcbarcode.com/news/rfid-and-inventory-control/>>, 14 Juni 2016.
- [19] Github, *RFID-RC522 Pin Out*, <URL: <https://github.com/r00tGER/RFID-RC522>>, 14 Juni 2016.
- [20] Sunrom, *RFID Reader/Writer 13.56MHz RC522 SPI*, <URL: <http://www.sunrom.com/p/mifare-rfid-readerwriter-1356mhz-rc522>>, 14 Juni 2016.

## DAFTAR RIWAYAT HIDUP



Nama : Muhammad Suprayogi  
TTL : Gresik, 17 April 1995  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat : Jl. Marabahan V/15 GKB  
Gresik  
Telp/HP : 089621877670  
E-mail : muh.suprayogi@gmail.com

### RIWAYAT PENDIDIKAN

1. 2001 – 2007 : MINU Trate Putera Unggulan Gresik
2. 2007 – 2010 : SMP Negeri 1 Gresik
3. 2010 – 2013 : SMA Negeri 1 Kebomas Gresik
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Komputer Kontrol - FTI Institut Teknologi Sepuluh Nopember (ITS)

### PENGALAMAN KERJA

1. Kerja Praktek di PT. IPMOMI Paiton Probolinggo

### PENGALAMAN ORGANISASI

1. Staff Dewan Perwakilan Mahasiswa Fakultas (2014-2015)
2. Kabiro Komunitas Dan Kesenian Divisi Minat Dan Bakat HIMAD3TEKTRO (2015-2016)

-----Halaman ini sengaja dikosongkan-----

## DAFTAR RIWAYAT HIDUP



Nama : Achmad Nurhisyam  
TTL : Malang, 26 April 1995  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat : Desa Wangkal RT. 10  
RW. 03 Probolinggo  
Telp/HP : 085646820234  
E-mail : achmad.nurhisyam@gmail.com

### RIWAYAT PENDIDIKAN

1. 2000 – 2001 : TK Kartini Wangkal
2. 2001 – 2007 : SDN Wangkal 02
3. 2007 – 2010 : SMP Negeri 1 Kraksaan
4. 2010 – 2013 : SMA Negeri 1 Kraksaan
5. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Komputer Kontrol - FTI Institut Teknologi Sepuluh Nopember (ITS)

### PENGALAMAN KERJA

1. Kerja Praktek di PT. IPMOMI Paiton Probolinggo

### PENGALAMAN ORGANISASI

1. Staff Divisi Big Event HIMAD3TEKTRO (2014-2015)
2. Staff BPM JMMI ITS (2014-2015)
3. Kabiro IARC Divisi Big Event HIMAD3TEKTRO (2015-2016)