

TUGAS AKHIR - TE145561

**PERANCANGAN DIGITAL *WIRELESS REMOTE STICK COMMANDER* UNTUK PENGENDALI *CAMERA CRANE* DAN *PAN TILT HEAD* BERBASIS *SENSOR ACCELEROGYRO***

Nova Agustina  
NRP 2214030013

Dosen Pembimbing  
Ir. Rusdhianto Efendi AK, MT

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 145561**

**PERANCANGAN DIGITAL WIRELESS REMOTE STICK  
COMMANDER UNTUK PENGENDALI CAMERA CRANE DAN  
PAN TILT HEAD BERBASIS SENSOR ACCELEROGYRO**

Nova Agustina  
NRP 2214030013

Dosen Pembimbing  
Ir. Rusdhianto Efendi AK, MT

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT - TE 145561**

**DESIGN OF WIRELESS REMOTE *STICK COMMANDER FOR CAMERA CRANE AND PAN TILT HEAD BASE ON ACCELEROGYRO SENSOR***

Nova Agustina  
NRP 2214030013

Advisor 1  
Ir. Rusdhianto Efendi AK, MT

COMPUTER CONTROL STUDY PROGRAM  
Electrical and Automation Engineering Departement  
Vocational Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



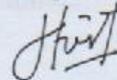
## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini kami menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir kami dengan judul "**Perancangan Digital Wireless Remote Stick Commander untuk Pengendali Camera Crane dan Pan Tilt Head Berbasis Sensor Accelerogyro**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang kami akui sebagai karya sendiri.

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

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

Surabaya, 20 Juli 2017



Nova Agustina  
NRP 2214030013

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

**PERANCANGAN DIGITAL *WIRELESS REMOTE STICK*  
COMMANDER UNTUK PENGENDALI *CAMERA CRANE* DAN  
*PAN TILT HEAD* BERBASIS SENSOR *ACCELEROGYRO***

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Program Studi Komputer Kontrol  
Departement Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing



Ir. Rusdhianto Efendi AK, MT

NIP. 1957042419850 2 1 001

**SURABAYA  
JULI, 2017**

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

# PERANCANGAN DIGITAL *WIRELESS REMOTE STICK COMMANDER* UNTUK PENGENDALI *CAMERA CRANE* DAN *PAN TILT HEAD* BERBASIS SENSOR *ACCELEROGYRO*

**Nama Mahasiswa** : Nova Agustina  
**NRP** : 2214030013  
**Pembimbing** : Ir. Rusdhianto Efendi AK, MT  
**NIP** : 19570424 198502 1 001

## ABSTRAK

Teknologi remot kontrol berkembang dengan pesat, remot digunakan hampir semua bidang. Pengendali jarak jauh atau remot kontrol adalah sebuah alat elektronik yang digunakan untuk mengoperasikan sebuah mesin atau *plant*. Salah satu contoh remot kontrol yang mudah digunakan adalah *stick commander*. *Stick commander* merupakan remot kontrol yang bentuknya seperti tongkat. Pada bidang multimedia *camera crane* masih dikendalikan secara konvensional atau dengan satu orang yang mengarahkan *camera crane* sesuai objek yang akan diambil dan didesain sedemikian rupa agar dapat dikendalikan dengan satu tangan dan dikendalikan dari jarak jauh

Pada Tugas Akhir ini dibuat *stick commander* berbasis sensor *accelero* dan *gyro* untuk mengendalikan *camera crane* dan *2 axis pan tilt head*. *Stick commander* dilengkapi dengan *joystick* untuk menggerakkan *pan tilt head* sedangkan sensor *accelero* dan *gyro* untuk mengendalikan pergerakan *camera crane* berdasarkan kemiringan *stick commander*. Ketika *stick commander* dalam keadaan aktif dapat menggerakkan *2 axis pan tilt head* tetapi saat *push button* di tekan LED berwarna hijau menyala dan *stick commander* dapat mengendalikan gerakan *camera crane* ke atas, bawah, kanan dan kiri. Dari hasil pengujian *stick commander* dapat mengendalikan *camera crane* dengan kemiringan *stick commander*. *Stick Commander* dapat mengendalikan *camera crane* dan *2 axis pan tilt head* hingga jarak 20 meter. Pengiriman data dengan komunikasi serial HC-12 berjalan dengan baik pada jarak hingga jarak 35 meter pada kondisi tanpa penghalang dengan delay menggunakan baud rate 9600.

**Kata Kunci** : *Stick Commander, Accelerogyro, Joystick, Camera crane, HC-12*

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

**DESIGN OF WIRELESS REMOTE STICK COMMANDER FOR  
CAMERA CRANE AND PAN TILT HEAD BASE ON  
ACCELEROGYRO SENSOR**

**Name** : Nova Agustina  
**NRP** : 2214030013  
**Advisor** : Ir. Rusdhianto Efendi AK, MT  
**NIP** : 19570424 198502 1 001

**ABSTRACT**

*Technology of remote control is growing rapidly, remote is used in almost every field. Remote control is an electronic device that used to operate a machine or plant. Example of an easy-to-use remote control is the stick commander. Stick commander is a remote control that looks like a stick. In the field of multimedia camera cranes are still controlled conventionally or with one person directing the camera crane according to the object to be taken and designed in such a way that can be controlled with one hand and controlled remotely*

*In this final project, made a stick commander based on accelero and gyro sensor to control camera crane and 2 axis pan tilt head. Stick commander is was completed with a joystick to move pan tilt head while accelero and gyro sensor to control the movement of camera crane based on slope stick commander. When the commander stick is activated it can move 2 axis pan tilt head but when push button on tap green LED is on and stick commander can control camera crane movement up, down, right and left. From the test results stick commander can control the camera crane with the slope stick commander. Stick Commander can contril camera crane and 2 axis pan tilt head to the 20 meter distance. Data shipping with serial HC-12 counsenly good with 35 meter distance at without backstop*

**Keywords :** *Stick Commander, Accelerogyro, Joystick, Camera crane, HC-12*

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

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Computer Control, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

***“PERANCANGAN DIGITAL WIRELESS REMOTE STICK  
COMMANDER UNTUK PENGENDALI CAMERA CRANE DAN  
PAN TILT HEAD BERBASIS SENSOR ACCELEROGYRO ”***

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Rusdhianto Efendi AK, MT. dan Bapak Ir. Joko Susila, MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, teman-teman angkatan 2014 yang banyak memberikan saran dan dukungan. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 20 Juli 2017

Penulis

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

# DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	<b>Error! Bookmark not defined.</b>
HALAMAN PENGESAHAN.....	ix
ABSTRAK.....	xi
<i>ABSTRACT</i> .....	xiii
KATA PENGANTAR .....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR .....	xxi
DAFTAR TABEL.....	xxiii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan .....	2
1.5 Metodologi Penelitian .....	2
1.6 Sistematika Laporan.....	3
1.7 Relevansi.....	4
<b>BAB II TEORI DASAR .....</b>	<b>5</b>
2.1 <i>Stick Commander</i> .....	5
2.2 <i>Sensor IMU (Inertial Measurement Unit)</i> .....	5
2.3 <i>Complementary Filter</i> .....	6
2.4 <i>Modul Joystick</i> .....	8
2.5 <i>Modul Boost Converter</i> .....	9
2.6 <i>Modul HC-12</i> .....	10
2.6.1 <i>Komunikasi Radio Frekuensi 433Mhz</i> .....	12
2.6.2 <i>Komunikasi UART ( Universal Asynchronous Receiver and Transmitter )</i> .....	12
2.7 <i>Batrai LiPO 3.7 Volt</i> .....	12
2.8 <i>Arduino Nano</i> .....	13
2.8.1 <i>Pemrograman Arduino IDE</i> .....	14
2.8.2 <i>ADC Arduino Nano</i> .....	15
2.8.3 <i>Input dan Output Arduino Nano</i> .....	15
2.9 <i>I<sup>2</sup>C (Inter Integrated Circuit)</i> .....	16

BAB III PERANCANGAN SISTEM <i>STICK COMMANDER</i> .....	19
3.1 Perancangan Hardware .....	19
3.1.1 Sensor IMU MPU6050 .....	20
3.1.2 Pengkabelan Arduino Nano dengan Sensor MPU6050 ...	21
3.1.3 Perancangan Catu Daya .....	22
3.1.4 Pengkabelan Arduino Nano dengan HC-12 .....	23
3.1.5 Pengkabelan Arduino Nano dengan Modul <i>Joystick</i> .....	25
3.1.6 Perancangan Rangkaian Push Button dan LED .....	26
3.1.7 Perancangan Rangkain <i>Shield</i> Arduino Nano .....	27
3.1.8 Perancangan Hardware (Perangkat Keras) <i>Stick Commander</i> .....	28
3.2 Perancangan <i>Software Stick Commander</i> .....	29
3.2.1 <i>Flowchart</i> .....	29
3.2.2 Pendeklarasian Variabel.....	32
3.2.3 Perancangan <i>Software Joystick</i> .....	32
3.2.4 Perancangan <i>Software Komunikasi Menggunakan HC-1233</i>	
3.2.5 Perancangan <i>Software Push Button</i> .....	34
3.2.6 Perancangan <i>Software Complementary filter</i> .....	35
BAB IV HASIL PERANCANGAN DAN PENGUJIAN.....	37
4.1 Pengujian Input/Output Arduino.....	37
4.2 Pengujian Power Supply dan <i>Boost Converter</i> .....	38
4.3 Pengujian Sensor IMU Tipe MPU6050 .....	39
4.4 Pengujian Modul <i>Joystick</i> .....	41
4.5 Pengujian Modul HC-12.....	42
4.6 Pengujian <i>Complementary Filter</i> .....	44
4.7 Pengujian <i>Stick Commander</i> terhadap <i>Camera Crane</i> dan <i>Pan Tilt Head</i> .....	47
BAB V PENUTUP .....	49
DAFTAR PUSTAKA .....	51
LAMPIRAN A .....	1
A.1 Listing Program <i>Stick Commander (Transmitter)</i> .....	1
A.3 Dokumentasi .....	6
LAMPIRAN B.....	1
B.1 Datasheet HC-12 .....	1
B.2 Datasheet <i>Joystick</i> .....	5
B.3 Datasheet MPU6050.....	6

B.4 Datasheet Bosst Conveter MT3608 .....	14
B.5 Datasheet Arduino Nano.....	18

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

## DAFTAR GAMBAR

HALAMAN

<b>Gambar 2.1</b> Tiga Sumbu Pesawat.....	5
<b>Gambar 2.2</b> Prinsip Kerja <i>Gyroscope</i> .....	6
<b>Gambar 2.3</b> Diagram Blok <i>Complementary Filter</i> .....	7
<b>Gambar 2.4</b> Modul <i>Joystick</i> .....	8
<b>Gambar 2.5</b> Skematik <i>Joystick</i> .....	9
<b>Gambar 2.6</b> Modul <i>Boost Converter</i> .....	9
<b>Gambar 2.7</b> Modul HC-12.....	10
<b>Gambar 2.8</b> Baterai LiPO 3,7 Volt .....	13
<b>Gambar 2.9</b> Arduino Nano .....	14
<b>Gambar 2.10</b> Pemrograman Arduino IDE .....	15
<b>Gambar 2.11</b> Sinyal ACK I <sup>2</sup> C .....	17
<b>Gambar 2.12</b> Trasfer <i>Bit</i> pada I <sup>2</sup> C <i>Bus</i> .....	17
<b>Gambar 3.1</b> Diagram Fungsional <i>Stick Commander</i> .....	19
<b>Gambar 3.2</b> Perancangan Elektrik Keseluruhan Sistem .....	20
<b>Gambar 3.3</b> Diagram Blok Arduino dengan MPU6050 .....	21
<b>Gambar 3.4</b> Konfigurasi MPU6050 dengan Arduino Nano .....	22
<b>Gambar 3.5</b> Diagram Blok Catu Daya.....	22
<b>Gambar 3.6</b> <i>Boost Converter</i> ke Arduino Nano .....	23
<b>Gambar 3.7</b> Diagram Blok Komunikasi .....	23
<b>Gambar 3.8</b> Konfigurasi HC-12 dengan Arduino Nano .....	24
<b>Gambar 3.9</b> Diagram Blok Arduino dengan joystick .....	25
<b>Gambar 3.10</b> Konfigurasi <i>Joystick</i> dengan Arduino Nano .....	25
<b>Gambar 3.11</b> Diagram Blok Led dan Push Button .....	26
<b>Gambar 3.12</b> Konfigurasi 2-Led Indikator dengan Arduino Nano.....	26
<b>Gambar 3.13</b> Rangkaian <i>Shield</i> Arduino Nano .....	27
<b>Gambar 3.14</b> Desain <i>Stick Commander</i> tampak Depan (a) dan Belakang (b).....	29
<b>Gambar 3.15</b> <i>Sub Program</i> .....	30
<b>Gambar 3.16</b> Flowchart Program <i>Stick Commander</i> .....	31
<b>Gambar 3.17</b> Deklarasi Variable .....	32
<b>Gambar 3.18</b> Program Fungsi <i>Joystick</i> .....	33
<b>Gambar 3.19</b> Program <i>Transmitter</i> Komunikasi Serial .....	34
<b>Gambar 3.20</b> Inialisasi Pin <i>Interrupt</i> .....	34

**Gambar 3.21** Blok Diagram Proses Complementary Filter..... 35  
**Gambar 3.22** Program *Complementary Filter*..... 36  
**Gambar 4.1** Cara Pengujian MPU6050 Sumbu-X (a) dan Sumbu Y (b) ..... 39  
**Gambar 4.2** Grafik pengujian complementary filter saat diam pada sb X..... 45  
**Gambar 4.3** Grafik pengujian complementary filter saat bergerak pada sb Y ..... 46  
**Gambar 4.4** Grafik Pengujian Complementary Filter ..... 46

## DAFTAR TABEL

### HALAMAN

<b>Tabel 2.1</b> Perintah AT Command.....	11
<b>Tabel 2.2</b> Spesifikasi Arduino Nano.....	13
<b>Tabel 3.1</b> <i>Mapping</i> pin analog dan digital yang digunakan.....	27
<b>Tabel 4.1</b> Hasil Pengujian <i>Input Output</i> Arduino Nano .....	38
<b>Tabel 4.2</b> Hasil Pengujian <i>Boost Converter</i> .....	38
<b>Tabel 4.3</b> Hasil Pengujian <i>Power Supply</i> .....	39
<b>Tabel 4.4</b> Hasil P engujian MPU6050 pada <i>Roll</i> , .....	40
<b>Tabel 4.5</b> Pengujian MPU6050 pada <i>Pitch</i> .....	40
<b>Tabel 4.6</b> Pengujian Joystick.....	42
<b>Tabel 4.7</b> Hasil Pengujian Respon <i>Joystick</i> .....	42
<b>Tabel 4.8</b> Pengujian <i>channel</i> berbeda.....	43
<b>Tabel 4.9</b> Pengujian <i>channel</i> sama .....	43
<b>Tabel 4.10</b> Pengujian dengan 3 <i>Baud Rate</i> Berbeda.....	43
<b>Tabel 4.11</b> Pengujian pada <i>Baud Rate</i> 9600 dengan Delay 0.1s .....	43
<b>Tabel 4.12</b> Pengujian Terdapat Halangan dengan <i>Baud Rate</i> dengan Delay 0.1 s .....	44
<b>Tabel 4.13</b> Hasil Pengujian camera crane dan pan tilt terhadap jarak ...	47
<b>Tabel 4.14</b> Hasil pengujian Respon pergerakan <i>roll</i> terhadap jarak .....	48

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

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Teknologi remot kontrol berkembang dengan pesat, remot digunakan hampir semua bidang. Pengendali jarak jauh atau remot kontrol adalah sebuah alat elektronik yang digunakan untuk mengoperasikan sebuah mesin dari jarak jauh.

Remot kontrol dibuat sedemikian rupa untuk menyesuaikan kebutuhan *plant* agar lebih mudah dan dapat dikendalikan dengan satu tangan. Pada bidang multimedia *camera crane* masih dikendalikan secara konvensional atau dengan satu orang yang mengarahkan *crane* sesuai objek yang akan diambil. Putusnya kabel yang menghubungkan *camera crane* dengan *pan-tilt* menjadi salah satu hal yang paling sering terjadi ketika pemasangan dan *set up camera crane*. Oleh karena itu diperlukan pengendali kamera *crane* dengan menggunakan satu tangan.

Pada Tugas Akhir ini akan dilakukan pembuatan *wireless* remot atau *stick commander* untuk mengendalikan *camera crane* dan *2 axis pan-tilt head*. *Stick commander* yang dilengkapi *joystick* dibuat untuk mengendalikan *2 axis pan-tilt head* dengan gerakan pada sumbu X dan Y. Sensor *accelero* dan *gyroscope* untuk mengendalikan *camera crane* sesuai dengan gerakan tangan, atau sudut yang diberikan oleh remot. Perangkat keras ini menggunakan Arduino sebagai mikrokontroler dan memprogram melalui Arduino IDE. Untuk menggerakkan *2 axis pan-tilt head* menggunakan *joystick*. Hasil yang di harapkan dengan adanya tugas akhir ini, dapat membuat *stick commander* untuk mengendalikan *camera crane* dan *2 axis pan-tilt head* secara *wireless* dari jarak jauh kurang lebih 15 meter.

### 1.2 Permasalahan

*Camera crane* dan *2 axis pan-tilt head* masih digunakan secara konvensional digerakkan oleh dua tangan. Untuk mempermudah juru kamera untuk menggerakkan *camera crane* dan *2 axis pan-tilt head* dengan satu tangan dan mengurangi penggunaan kabel. Sehingga diperlukan pengendali untuk mempermudah pergerakan *camera crane* dan *2 pan-tilt head* dengan satu orang secara *wireless*.

### 1.3 Batasan Masalah

Dalam perancangan dan pembuatan Tugas Akhir ini, penulis membuat batasan dalam pengerjaan, yaitu:

- a) *Stick Commander* masih diaplikasikan pada *camera crane* dan *2 axis pan tilt head*.
- b) Jarak maksimal antara *stick commander* dengan *camera crane* adalah 15 meter
- c) Filter yang digunakan untuk menggabungkan sensor *accelerometer* dan *gyroscope* adalah *complementary filter*.

### 1.4 Tujuan

Pembuatan sistem kendali *camera crane* dan *2 axis pan-tilt head* menggunakan *stick commander* ini dapat membuat dan merancang *stick commander* untuk menggerakkan *camera crane* berdasarkan sudut *stick commander* serta menggerakkan *pan tilt* dengan *joystick* secara wireless dengan satu tangan.

### 1.5 Metodologi Penelitian

Dalam pelaksanaan Tugas Akhir yang berupa “Perancangan Digital *Wireless Remote Stick Commander* untuk Pengendali *Camera Crane* dan *Pan Tilt Head* Berbasis Sensor *Accelerogyro*”, ada beberapa kegiatan yang dapat diuraikan sebagai berikut:

#### a) Studi Pustaka dan Survei Data Awal

Kegiatan ini dilakukan dengan mengidentifikasi masalah yang akan dibahas dan mempelajari buku pedoman dari *e-book* atau jurnal yang menunjang teori atau prinsip dasar rangkaian, baik *software* maupun *hardware* dalam pembuatan alat tersebut. Selain itu, dalam kegiatan ini juga dilakukan beberapa survei lapangan atau analisa kondisi lingkungan, guna mengetahui seberapa besar alat ini mempunyai nilai kebermanfaatan jika direalisasikan dalam kehidupan sehari-hari.

#### b) Perancangan dan Pembuatan Alat

Pada tahapan ini dilakukan perancangan dan pemodelan sistem *stick commander* mulai dari bagian elektriknya sampai bagian mekaniknya. Dan alat akan didesain sedemikian rupa agar nantinya dapat direalisasikan dalam kehidupan sehari-hari. Setelah melakukan perancangan bentuk, desain dan struktur dari sistem, barulah dilakukan pembuatan *hardware* dari sistem tersebut. Yang meliputi pembuatan sistem mekanik dan sistem elektroniknya.

c) **Perancangan dan Pembuatan Perangkat Lunak**

Perancangan dan pembuatan *software* dilakukan pada sistem minimum mikrokontroller ATmega328 menggunakan bahasa pemrograman IDE.

d) **Uji Coba dan Analisis Data**

Pengujian ini dimaksudkan untuk memastikan bahwa kinerja masing-masing komponen dari hasil pembuatan alat dapat berfungsi sesuai yang diharapkan. Untuk proses pengujian sistem *stick commander* dilakukan melalui percobaan dengan menggunakan *camera crane*. Dari pengujian ini juga dilakukan penganalisaan sistem untuk memperoleh data-data sistem, seberapa besar nilai *error* pada sistem.

e) **Penyusunan Laporan**

Penyusunan laporan dilakukan setelah semua tahap terselesaikan sehingga hasil yang diperoleh dari pembuatan alat dapat dijelaskan secara rinci dan spesifik sesuai dengan data-data yang diperoleh.

## 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 *stick commander*, sensor IMU (*accelerometer dan gyroscope*), *Complementary Filter*, *Joystick*, *Komunikasi data UART*, *Boost Converter*, Modul HC-12, Arduino Nano dan I2C.

**Bab III    Perancangan Sistem**

Bab ini membahas pembuatan *stick commander* 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 memberi manfaat berikut:

- a) Mempermudah juru kamera, karena juru kamera tidak menggerakkan camera crane secara manual memindahkan arah *crane* sesuai dengan pergerakan objek yang akan diambil. Dengan menggunakan *stick commander* juru kamera dapat menggerakkan *crane* dan *pan-tilt head* sekaligus secara *wireless*
- b) Mengembangkan remot kontrol atau *stick commander*

## BAB II TEORI DASAR

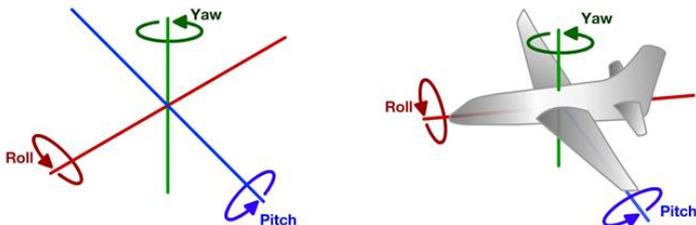
Pada bab ini akan dibahas mengenai teori penunjang dari peralatan dan teori yang digunakan dalam sistem *stick commander* seperti Sensor IMU, *complementary filter*, modul *joystick*, modul *boost converter*, modul HC-12, Arduino Nano dan I2C (*Inter Integrated Circuit*).

### 2.1 *Stick Commander*

*Stick commander* merupakan remot kontrol yang bentuknya seperti tongkat. *Stick commander* digunakan untuk mengendalikan atau mengomandakan sebuah peralat dari jarak jauh. *Stick commander* didesain sedemikian rupa agar dapat dikendalikan dengan satu tangan dan dikendalikan dari jarak jauh. *Stick commander* yang dibuat untuk menggerakkan *camera crane* dan *2 axis pan tilt head* dibuat dilengkapi dengan sensor *accelerogyro* dan *joystick*.

### 2.2 Sensor IMU (*Inertial Measurement Unit*)

Sebuah sensor yang digunakan untuk menentukan atau mengukur sudut *roll*, sudut *pitch* dan sudut *yaw*. Ketiga sudut ini menentukan orientasi atau *attitude*, dari sebuah pesawat terbang atau benda yang bergerak relatif terhadap bumi, terlihat pada Gambar 2.1. Sensor ini mengumpulkan data kecepatan angular dan akselerasi linear yang kemudian dikirim ke *Central Processing Unit* (CPU) untuk mendapatkan data keberadaan dan pergerakan suatu benda. IMU terdiri dari kombinasi *accelerometer* (sensor percepatan) dan *gyroscope* (sensor kecepatan angular).

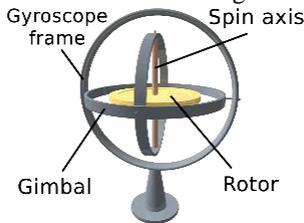


**Gambar 2.1** Tiga Sumbu Pesawat

*Accelerometer* adalah sensor yang digunakan untuk mengukur percepatan suatu objek. *Accelerometer* mengukur percepatan dinamik dan statik. Pengukuran dinamik adalah pengukuran percepatan pada objek bergerak, sedangkan pengukuran statik adalah pengukuran terhadap gravitasi bumi. *Accelerometer* tidak bisa mengikuti pergerakan yang cepat dikarenakan responnya yang lamban. *Accelerometer* dapat memberikan pengukuran sudut tegak lurus yang akurat ketika sistem sedang diam (statis). Untuk mengubah data *accelerometer* menjadi sudut menggunakan Persamaan (2.2).

$$\left. \begin{aligned} x &= \frac{180}{\pi} \times \arctan\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \\ y &= \frac{180}{\pi} \times \arctan\left(\frac{x}{\sqrt{y^2 + z^2}}\right) \\ z &= \frac{180}{\pi} \times \arctan\left(\frac{\sqrt{x^2 + y^2}}{z}\right) \end{aligned} \right\} \quad (2.2)$$

*Gyroscope* adalah perangkat untuk mengukur atau mempertahankan orientasi, dengan prinsip ketetapan momentum sudut. Mekanismenya adalah sebuah roda berputar dengan piringan didalamnya yang tetap stabil. *Gyroscope* sering digunakan pada robot atau helikopter dan alat-alat canggih lainnya. *Gyroscope* menentukan orientasi gerak dengan bertumpu pada roda atau cakram yang berotasi dengan cepat pada sumbu. *Gyro* sensor bisa mendeteksi gerakan sesuai gravitasi, atau dengan kata lain mendeteksi gerakan pengguna.



**Gambar 2.2** Prinsip Kerja *Gyroscope*

### 2.3 Complementary Filter

*Accelerometer* dapat memberikan pengukuran sudut kemiringan (*tilt*) yang akurat ketika sistem sedang diam (*statis*). Bila sistem sedang berotasi atau bergerak, *accelerometer* tidak bisa mengikuti pergerakan

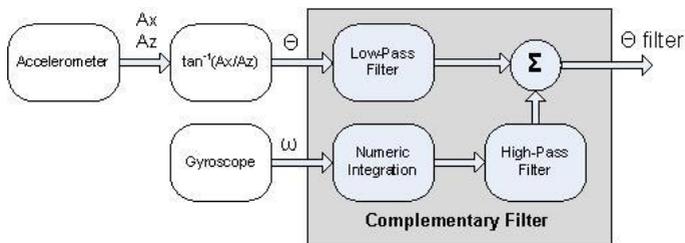
yang cepat dikarenakan responnya yang lambat dan memiliki *noise*. *Gyroscope* dapat membaca kecepatan sudut (*angular rate*) yang dinamis. Setelah melakukan komputasi menggunakan integral data dari waktu ke waktu, maka perpindahan sudut atau sudut kemiringan dapat dihitung. Tetapi sudut ini akan menjadi tidak akurat dalam jangka panjang karena efek *bias* yang dihasilkan oleh *gyroscope*.

Salah satu cara paling sederhana untuk menggabungkan data dari *gyro* dan *accelerometer* adalah dengan menggunakan *Complementary Filter*. *Complementary Filter* dirancang sedemikian rupa sehingga kelebihan dari satu sensor akan digunakan untuk mengatasi kekurangan dari sensor lain yang melengkapi satu sama lain. Hal ini akan menghasilkan estimasi bebas dari penyimpangan sudut dan cepat menanggapi pergeseran. Gambar 2.3 menunjukkan bagaimana *gyro* dan *accelerometer* digabungkan bersama.

Menggunakan 2 sumbu sensor *accelerometer* untuk mendapatkan nilai kemiringan sudut ( $\theta$ ), yaitu sumbu X dan Z. Kemudian sudut kemiringan dari *accelerometer* ini diumpankan ke *low-pass filter* untuk menghilangkan *noise*. Sedangkan keluaran sensor *gyroscope* yang berupa kecepatan sudut ( $\omega$ ) harus diintegrasikan terlebih dahulu untuk mendapatkan nilai perpindahan sudut atau sudut kemiringan. Kemudian diumpankan ke *high-pass filter* untuk menghilangkan efek bias pada *gyroscope*.

Jika *complementary filter* ini diaplikasikan ke bahasa pemrograman, maka akan menjadi Persamaan (2.3).

$$sudut = (a) \times (sudut + gyro \times dt) + (a - 1) \times (acc) \quad (2.3)$$



**Gambar 2.3** Diagram Blok *Complementary Filter*

Dimana:

$a$  : koefisien filter

$dt$  : waktu sampling, disesuaikan dengan waktu sampling sensor

Sudut : sudut keluaran *complementary filter*

*gyro* : keluaran sensor *gyroscope* berupa kecepatan sudut  
*acc* : keluaran sensor *accelerometer* yang sudah berupa sudut  
*time constant* ( $\tau$ ) dapat ditentukan dengan Persaman (2.4)

$$\tau = (a \times dt) / (1 - a) \quad (2.4)$$

*Time constant* adalah lamanya waktu update sinyal keluaran dari *complementary filter*. Semakin besar nilai koefisien filter ( $a$ ) maka sinyal keluaran *complementary filter* akan lebih stabil (semakin tidak ber-*noise*) tetapi akan memperlama *time constant* yang mengakibatkan waktu *update* sinyal semakin lama. Semakin cepat waktu *sampling* yang digunakan maka akan semakin cepat *time constant* dari sinyal keluaran *complementary filter*. Pemilihan nilai koefisien filter ( $a$ ) dan waktu *sampling* ( $dt$ ) harus disesuaikan dengan respon pada aplikasi yang dibuat. Jika respon sistem aplikasi anda cepat, maka disarankan untuk mempercepat nilai *time constant*, dan jika respon sistem pada aplikasi yang anda buat relatif lambat, maka nilai *time constants* pada algoritma *complementary filter* dapat anda buat relatif lebih lama.

## 2.4 Modul Joystick

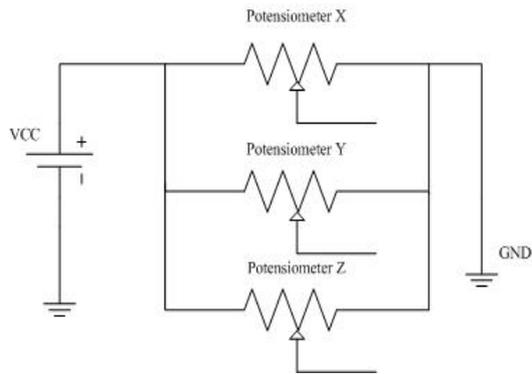
Modul *joystick* seperti Gambar 2.4 ini sama seperti *joystick* analog pada *gamepad*. Ini dibuat dengan memasang 2 buah potensiometer dengan sudut  $90^\circ$ . Potensio terhubung dengan *stick* pendek yang berpusat pada pegas.



**Gambar 2.4** Modul *Joystick*

Modul ini menghasilkan *output* sekitar 2.5 volt dari X dan Y bila dalam posisi ditengah. Menggerakkan *joystick* akan menyebabkan output yang bervariasi dari 0 volt hingga 5 volt tergantung pada arah yang ditentukan. Jika modul dihubungkan dengan mikrokontroler maka dapat dilihat nilainya sebsar 512 bila berada pada posisi di tengah. Bila *joystick* digerakkan maka nilainya akan beurbah antara 0-1023

tergantung pada posisinya. Diagram skematiknya terlihat pada Gambar 2.5



**Gambar 2.5** Skematik *Joystick*

## 2.5 Modul *Boost Converter*

*Boost converter* yaitu menaikkan tegangan dengan level output bervariasi dan level input tetap. Tegangan input mempunyai range tertentu. Komponen utama dari *boost converter* adalah kumparan, diode, kapasitor, dan rangkaian *clock generator* dengan frekuensi tertentu. *boost converter* mempunyai efisiensi tinggi, sehingga tidak banyak panas yang dihasilkan dari sistem ini. Modul *boost converter* sangat cocok untuk menaikkan tegangan baterai 3.7 volt, kemudian tegangan kita atur 5 Volt untuk digunakan sebagai *supply* Arduino. Modul *boost converter* dapat dilihat pada Gambar 2.6[6]



**Gambar 2.6** Modul *Boost Converter*

Sebagaimana *buck-converter*, *boost-converter* juga menerapkan sistem SMPS (*Switching Mode Power Supply*), maka ia adalah bagian dari jenis power-supply SMPS. Memiliki efisiensinya tinggi. Adapun level tegangan keluaran yang dapat dihasilkan oleh *boost-converter* secara praktis didapatkan dengan perhitungan :

$$V_O = \frac{V_{in}}{1-D} \dots \dots \dots (2.1)$$

## 2.6 Modul HC-12

Modul HC-12 adalah modul komunikasi *serial half-duplex nirkabel* dengan 100 saluran di kisaran 433,4-473,0 MHz yang mampu mentransmisikan hingga 1 km. HC-12 dilengkapi dengan antena untuk memancarkan sinyal frekuensi radio. *Transceiver* HC-12, yang menyediakan antarmuka UART 4-pin TTL (Vcc, Gnd, Tx, Rx), dengan pin ke-5 yang digunakan untuk memasuki mode "perintah" Untuk mengubah konfigurasi modul HC-12 memiliki 100 saluran yang didukung berjarak 400 kHz, delapan tingkat transmisi, delapan tingkat baud yang didukung, dan tiga mode kerja yang berbeda[7]. Modul HC-12 dapat dilihat pada Gambar 2.7

Spesifikasi HC-12

Memiliki 5 pin kaki	: (VCC, GND, RX, TX, SET )
Berkerja pada frekuensi	: 433.4 Mhz – 473.0 Mhz
Tegangan <i>supply</i>	: 3.2V-5.5V
Jarak komunikasi	: 1000m di ruangan terbuka
<i>Serial baudrate</i>	: 1.2Kbps – 115.2 (default 9,6 Kbps)
<i>Receiving sensitivity</i>	: -117dBm hingga 100dBm
<i>Transmit power</i>	: -1dBm hingga 20dBm
<i>Interface Protokol</i>	: UART/TTL
<i>Operating Temperature</i>	: -40°C hingga +85°C
Daya Maksimum Transmisi	: 100mW (20dBm)
Dimensi	: 27.8mm × 14.4mm × 4mm



**Gambar 2.7** Modul HC-12

Pengaturan umum pada HC-12 adalah pengaturan AT *Command*. AT *Command* berfungsi untuk melakukan konfigurasi dari modul HC-12. Pengaturan AT *Command* memiliki beberapa perintah antara lain, dapat dilihat pada Tabel 2.1:

**Tabel 2.1** Perintah AT Command

No.	Perintah	Kegunaan
1.	AT	Untuk menguji apakah modul telah siap digunakan. Apabila respon dari perintah tersebut tertulis “OK” pada <i>serial monitor</i> maka modul HC-12 siap digunakan
2.	AT-Bxxx	Digunakan untuk mengatur nilai <i>Baud Rate</i> sebagai berikut : 1200bps, 2400bps, 4800bps, 9600bps, 19.200bps, 38.400bps, 57.600bps, atau 115.200bps.
3.	AT-Cxxx	Untuk saluran komunikasi nirkabel, pilih dari 001 sampai 127. Itu nilai <i>default</i> untuk saluran nirkabel adalah 001, dengan frekuensi kerja 433,4MHz. Saluran loncatan adalah 400KHz, dan frekuensi kerja saluran 100 adalah 473.0MHz.
4.	AT-Fux	Untuk mengubah modus transmisi <i>port serial</i> dari modul. Empat mode tersedia, yaitu FU1, FU2, FU3, dan FU4
5.	AT-DEFAULT	Untuk mengatur port serial <i>baud rate</i> dan konfigurasi, saluran komunikasi, <i>power</i> , dan mode transmisi <i>port serial</i> transparan kembali ke awal
6.	AT-Px	Untuk mengatur nilai dari daya pemancar modul. Terdapat nilai 1-8 dengan daya yang berbeda beda
7.	AT-Udbs	Menetapkan bit data (d), paritas (p), dan <i>stop bit</i> untuk komunikasi <i>port serial</i> untuk paritas.
8.	AT-Ry	Menentukan satu parameter dari modul, di mana y adalah huruf antara B, C, F, dan P, masing-masing mewakili: <i>baud rate</i> , saluran komunikasi, port serial, mode transmisi transparan, dan daya transmisi.
9.	AT-Rx	Dapatkan semua parameter dari modul mengembalikan <i>port serial</i> secara transparan, mode transmisi, <i>baud rate port serial</i> , saluran komunikasi, dan mentransmisikan

### **2.6.1 Komunikasi Radio Frekuensi 433Mhz**

Radio frekuensi merupakan gelombang elektromagnetik yang digunakan oleh sistem komunikasi untuk mengirim informasi melalui udara dari satu titik ke titik lain. Dikarenakan karekteristik bervariasi, gelombang radio dari frekuensi yang berbeda-beda digunakan tidak hanya untuk *broadcasting* tetapi juga pada perangkat tanpa kabel, transmisi telepon, televisi, radar, sistem navigasi, dan bentuk lain dari komunikasi yang mirip seperti sistem telemetri. Sinyal radio frekuensi merambat diantara antena pemancar pengirim dan penerima. Sinyal yang dipasok pada antena memiliki amplitudo, frekuensi dan interval. Hal tersebut berubah-ubah setiap saat merepresentasikan informasi.

### **2.6.2 Komunikasi UART ( *Universal Asynchronous Receiver and Transmitter* )**

Komunikasi serial adalah pengiriman data secara serial (data dikirim satu persatu secara berurutan), sehingga komunikasi serial jauh lebih lambat daripada komunikasi paralel. Komunikasi serial ada dua macam, *asynchronous serial* dan *synchronous serial*. Untuk komunikasi UART menggunakan komunikasi serial *asynchronous*. *Asynchronous serial* adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan *clock* namun hanya data yang ditransmisikan, tanpa *clock*. Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi *clock* harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi *clock* pengirim dan penerima akan membaca data sesuai dengan frekuensi *clock* penerima. Contoh penggunaan *asynchronous serial* adalah pada *Universal Asynchronous Receiver Transmitter* (UART) yang digunakan pada serial port (COM) komputer.

### **2.7 Batrai LiPO 3.7 Volt**

Baterai *Lithium Polymer* merupakan salah satu dari beberapa baterai yang berjenis *Lithium* yang ada di dunia. Baterai ini berbeda dengan *lithium-ion* (yang digunakan di ponsel) diantaranya kerapatan energi, arus, *charge rate*, dan tagangan yang sangat tinggi di banding dengan batrai yang berjenis lain. Baterai *Lithium Polymer* ini mempunyai tegangan normal 3.7 volt akan tetapi ketika baterai diisi

penyimpanan akan mempunyai tegangan 4.2 volt. Ketika sedang melakukan pengisian ulang tegangannya tidak boleh melebihi 4.2 volt, oleh karena itu baterai jenis Lipo membutuhkan *charger* baterai lipo khusus yang sudah dilengkapi dengan *auto cutoff* dan *ballance charger*.



**Gambar 2.8** Baterai LiPO 3,7 Volt

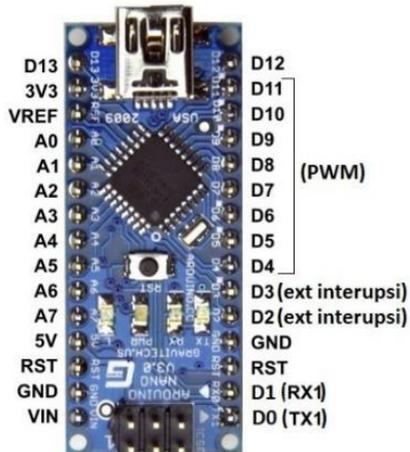
## 2.8 Arduino Nano

Arduino merupakan *prototyping platform* yang *open-source* dibuat berdasarkan kemudahan pemakaian perangkat keras dan perangkat lunak. Papan arduino dapat digunakan membaca input dan mengubahnya menjadi sebuah *output* (mengaktifkan motor, menyalakan sebuah LED). Arduino Nano adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan *breadboard*. Arduino Nano diciptakan dengan basis mikrokontroler ATmega328 (untuk Arduino Nano versi 3.x) atau ATmega 168 (untuk Arduino versi 2.x). Arduino Nano tidak menyertakan colokan DC, dan dihubungkan ke komputer menggunakan *port* USB Mini-B[6]. *Board* Arduino Nano dapat dilihat pada Gambar 2.9

**Tabel 2.2** Spesifikasi Arduino Nano

Mikrokontroler	Atmega328
Tegangan Pengoperasian	5 Volt
Batas tegangan yang disarankan	7-12 Volt
Batas tegangan <i>Input</i>	6-20 Volt
Jumlah pin I/O <i>Digital</i>	14 <i>pin digital</i> (6 diantaranya menyediakan keluaran PWM)
Jumlah pin <i>input Analog</i>	8pin
Arus DC tiap Pin I/O	40mA
Arus DC tiap Pin 3.3V	50mA
<i>Memory flash</i>	32KB(Atmega 328) sekitar

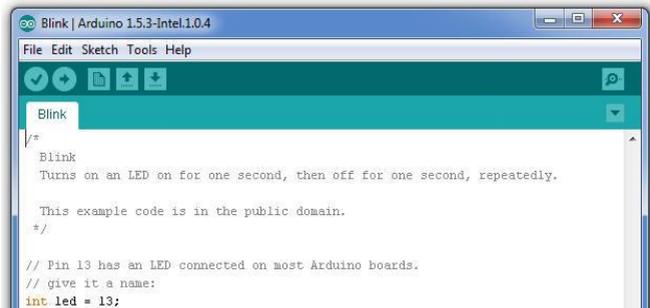
	0.5KB digunakan oleh <i>boodloader</i>
SRAM	2KB (ATmega 328)
EPROM	1KB (Atmega 328)
<i>Clockspeed</i>	16Mhz



**Gambar 2.9** Arduino Nano

### 2.8.1 Pemrograman Arduino IDE

Untuk memprogram Arduino menggunakan pemrograman Arduino IDE. IDE adalah sebuah *software* untuk menulis program, meng-compile menjadi kode biner dan meng-*upload* ke dalam memori mikrokontroller. Arduino berevolusi menjadi sebuah *platform* karena ia menjadi pilihan dan acuan bagi banyak praktisi. Salah satu yang membuat Arduino memikat hati banyak orang adalah karena sifatnya yang *open source*, baik untuk *hardware* maupun *software* nya. Desain *hardware*, bahasa pemrograman dan IDE Arduino yang berkualitas tinggi dan sangat berkelas mestinya menjadi hal yang dapat diandalkan dalam dunia IT. Saat ini *software* Arduino yang digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan Arduino.



**Gambar 2.10** Pemrograman Arduino IDE

### 2.8.2 ADC Arduino Nano

*Analogue to Digital Converter* (ADC) adalah fitur yang sangat berguna yang mengubah tegangan analog pada pin ke nomor digital. Dengan beralih dari sinyal analog ke data digital. Tidak setiap pin pada mikrokontroler memiliki kemampuan melakukan konversi analog ke digital. Pada papan Arduino, pin ini memiliki 'A' di depan labelnya (A0 sampai A5) untuk menunjukkan pin ini dapat membaca tegangan analog. ADC dapat sangat bervariasi antara mikrokontroler. ADC pada Arduino adalah ADC 10 bit yang artinya memiliki kemampuan untuk mendeteksi 1.024 ( $2^{10}$ ) tingkat analog diskrit. Beberapa mikrokontroler memiliki ADC 8 bit ( $2^8 = 256$  tingkat diskrit) dan beberapa memiliki ADC 16 bit ( $2^{16} = 65.536$  tingkat diskrit).

Cara kerja ADC cukup kompleks. Ada beberapa cara yang berbeda untuk mencapai prestasi ini, namun salah satu teknik yang paling umum menggunakan tegangan analog untuk mengisi sebuah kapasitor internal dan kemudian mengukur waktu yang diperlukan untuk melepaskan resistor internal. Mikrokontroler memonitor jumlah siklus *clock* yang dilewati sebelum kapasitor dilepaskan.

### 2.8.3 Input dan Output Arduino Nano

Arduino Nano memiliki 14 buah pin digital yang dapat digunakan sebagai input atau output, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus 20mA, dan memiliki tahanan *pull-up* sekitar 20-50k ohm (secara *default* dalam posisi *disconnect*). Nilai maximum adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan *chip* mikrokontroler.

Beberapa pin memiliki fungsi khusus yaitu Serial, terdiri dari 2 pin pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial. *External Interrupts*, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan interrupts. Gunakan fungsi *attachInterrupt()*. PWM: Pin 3, 5, 6, 9, 10, dan 11 menyediakan output PWM 8-bit dengan menggunakan fungsi *analogWrite()*. SPI : Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan SPI Library. LED : Pin 13. Pada pin 13 terhubung *built-in* LED yang dikendalikan oleh digital pin no 13.

Arduino Nano memiliki 8 buah *input* analog, yang diberi tanda dengan A0 hingga A7. Masing-masing pin analog tersebut memiliki resolusi 1024 bits. Secara *default*, pin-pin tersebut diukur dari *ground* ke 5V, namun bisa juga menggunakan pin REF dengan menggunakan fungsi *analogReference()*. Pin Analog A6 dan A7 tidak bisa dijadikan sebagai pin digital, hanya sebagai analog. Beberapa pin lainnya pada board ini adalah. I2C : Pin A4 (SDA) dan A5 (SCL). Pin ini mendukung komunikasi I2C (TWI) dengan menggunakan Wire Library. AREF.

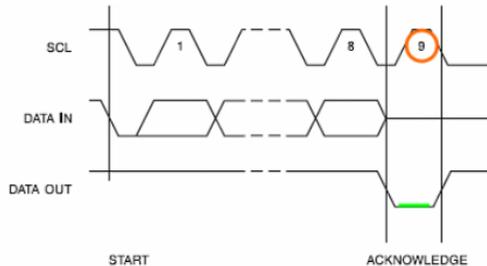
## 2.9 I<sup>2</sup>C (*Inter Integrated Circuit*)<sup>[11]</sup>

*Inter Integrated Circuit* atau sering disebut I<sup>2</sup>C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Sistem I<sup>2</sup>C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I<sup>2</sup>C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai *transfer* data pada I<sup>2</sup>C Bus dengan membentuk sinyal *Start*, mengakhiri *transfer* data dengan membentuk sinyal *Stop*, dan membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati *master*.

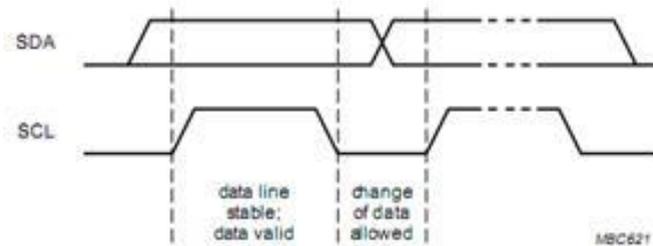
Sinyal *Start* merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “1” menjadi “0” pada saat SCL “1”. Sinyal *Stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1”. Kondisi sinyal *Start* dan sinyal *Stop* seperti tampak pada Gambar 2.11

Sinyal dasar yang lain dalam I<sup>2</sup>C 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 tampak pada Gambar 2.11.



**Gambar 2.11** Sinyal ACK I<sup>2</sup>C



**Gambar 2.12** Trasfer *Bit* pada I<sup>2</sup>C *Bus*

Dalam melakukan transfer data pada I2C *Bus*, kita harus mengikuti tata cara yang telah ditetapkan yaitu pertama transfer data hanya dapat dilakukan ketika *Bus* tidak dalam keadaan sibuk. Kedua selama proses transfer data, keadaan data pada SDA harus stabil selama SCL dalam keadan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal *start* atau sinyal *stop*. Transfer bit terlihat pada Gambar 2.12.

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

## BAB III

### PERANCANGAN SISTEM *STICK COMMANDER*

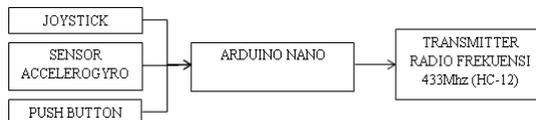
Dalam bab ini akan membahas mengenai perancangan alat yang meliputi perencanaan perangkat keras (*hardware*) dan perangkat lunak (*software*). Perancangan alat akan dibahas perbagian disertai dengan gambar skematik. Sedangkan penjelasan *software* akan dijelaskan mengenai pembuatan program di Arduino, serta program komunikasi serialnya menggunakan Arduino.

Untuk memudahkan dalam pembahasan bab ini akan dibagi menjadi tiga yaitu:

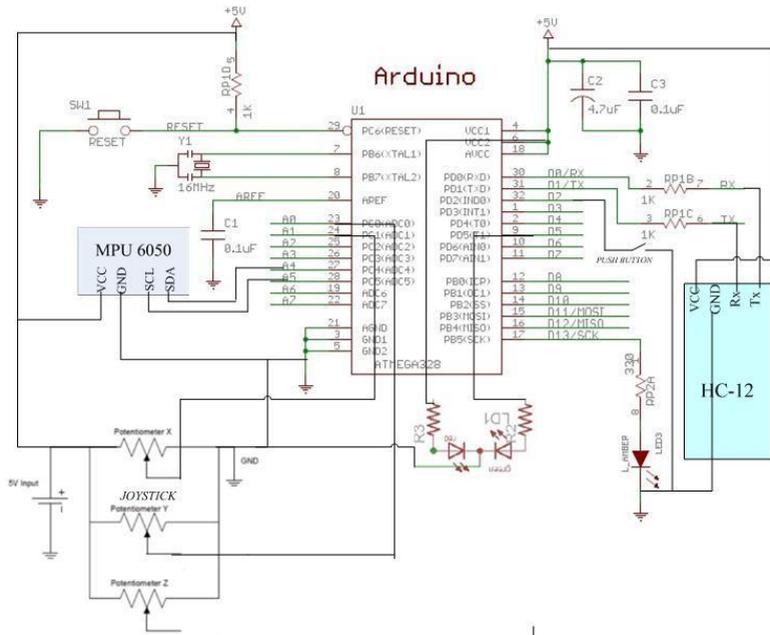
- a. Perancangan *hardware* (perangkat keras) yang terdiri dari konfigurasi pin Arduino, pembuatan rangkain *shield*, konfigurasi sensor *MPU6050*, konfigurasi *joystick*, konfigurasi *transmitter* menggunakan HC-12, pembuatan rangkain *push button*.
- b. Perancangan *software* (perangkat lunak) meliputi perancangan pembuatan program di Arduino .
- c. Perancangan *stick commander* meliputi desain perancangan *stick commander* yang digunakan untuk memasukan semua perangkat elektronik supaya terlihat lebih ringkas dan mudah digunakan. Bahan yang digunakan untuk mebuat *stick commander* ini menggunakan bahan *acrylic*.

#### 3.1 Perancangan Hardware

Secara garis besar cara kerja *stick commander* dapat dilihat dari diagram fungsional pada gambar 3.1. Sistem kerja *stick commander* secara keseluruhan yang pertama pengambilan nilai ADC *joystick*, yang kedua saat *push button* di tekan akan dilakukan pengambilan sudut kemiringan *stick commander* oleh sensor *accelerogyro*. Kemudian data ADC *joystick* dan kemiringan *stick commander* dikirimkan ke *reciver* menggunakan HC-12. Perancangan elektrik sistem keseluruhan dapat dilihat pada gambar 3.2



**Gambar 3.1** Diagram Fungsional *Stick Commander*



**Gambar 3.2** Perancangan Elektrik Keseluruhan Sistem

### 3.1.1 Sensor IMU MPU6050

Merupakan 6 axis Motion Processing Unit atau 6 degrees of freedom (DOF) yang berisi sensor accelerometer (3 DOF) dan gyroscope (3DOF) dan tambahan sensor suhu. Modul ini memiliki interface I2C yang di atur dengan kecepatan 400kbps. Sensor ini sangat akurat dengan fasilitas hardware internal 16 bit ADC untuk setiap kanalnya. Sensor ini akan menangkap nilai kanal axis X, Y dan Z bersamaan dalam satu waktu. Memiliki spesifikasi sebagai berikut :

- Gyroscope range + 250 500 1000 2000 ° / s
- Acceleration range:  $\pm 2 \pm 4 \pm 8 \pm 16$  g
- Communication standard I2C, address 0x68
- Chip built-in 16 bit AD converter, 16 bits data output
- Dimensi modul 20.3mm x 15.6mm

Pengaturan register pada MPU-6050 dilakukan oleh mikrokontroler dengan komunikasi I2C. Perintah yang digunakan

adalah perintah tulis. Dalam pengaturan register MPU-6050 perlu diperhatikan alamat dari MPU-6050 tersebut dan alamat register yang akan diatur. Alamat *default* dari MPU-6050 adalah 0x68, alamat ini dijadikan alamat untuk MPU-6050 pertama sebagai pemberi nilai set point. Sedangkan alamat dari MPU-6050 kedua sebagai pemberi nilai feedback adalah 0x69. Alamat pada MPU-6050 dapat diubah menjadi 0x69 dengan cara memberi tegangan 3.3V pada pin AD0.

Pembacaan *sensor gyroscope* pada MPU-6050 disimpan pada register data. Perangkat luar dapat meminta data tersebut dengan menunjuk alamat dari register data tersebut. Data *gyroscope* memiliki lebar data 16-bit yang terdiri dari 8-bit low byte dan 8-bit *high byte*. Terdapat tiga buah sumbu pada *gyroscope* sehingga terdapat enam buah register data *gyroscope*.

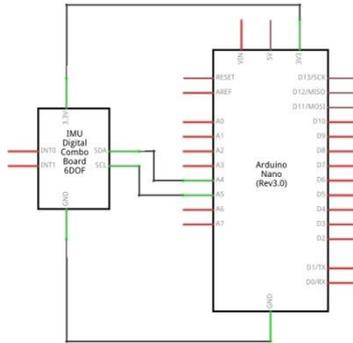
### 3.1.2 Pengkabelan Arduino Nano dengan Sensor MPU6050

Perencanaan Tugas Akhir ini menggunakan sensor MPU6050, merupakan sensor *IMU (Inertial Measurement Unit)*, yang berfungsi untuk membaca sudut yang diberikan pada *stick commander*. Pada blok diagram Gambar 3.3 merupakan hubungan antara sensor accelero dan gyro sebagai input. Sensor MPU 6050 menggunakan data serial yang dikirimkan ke arduino, berupa nilai accelerometer dan gyroscope



**Gambar 3.3** Diagram Blok Arduino dengan MPU6050

Terdapat 8 pin pada sensor MPU6050 yaitu VCC, GND, SCL, SDA, XCL, AD0 dan INT. Namun yang digunakan hanya VCC, GND, SCL dan SDA, pin SCL SDA digunakan untuk komunikasi I<sup>2</sup>C. Untuk pin SCL terhubung ke A5 dan SDA terhubung ke A4. Pin A4 (SDA) dan A5 (SCL). Pin ini mendukung komunikasi I2C (TWI) dengan menggunakan Wire Library. Pengkabelan MPU6050 dengan Arduino dapat dilihat pada gambar 3.4



**Gambar 3.4** Konfigurasi MPU6050 dengan Arduino Nano

### 3.1.3 Perancangan Catu Daya

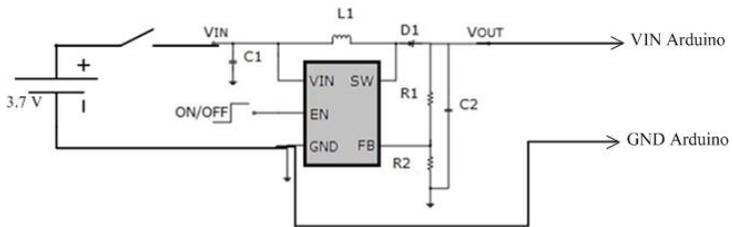
Perancangan catu daya pada Tugas Akhir ini menggunakan baterai *lippo* 3.7Volt 600mAh. Baterai tersebut digunakan sebagai catu daya pada Arduino, namun sebelum masuk ke Arduino masuk ke dalam rangkaian *boost converter* dapat dilihat pada gambar 3.3. Rangkaian tersebut digunakan untuk menaikkan tegangan dari 3,7 Volt ke 5 Volt. Prinsip utama untuk menaikkan tegangan adalah kemampuan induktor untuk menyimpan energi. Dalam *boost* konverter , tegangan *output* selalu lebih tinggi dari tegangan input . Berikut merupakan blok diagram *boost converter* pada Gambar 3.5



**Gambar 3.5** Diagram Blok Catu Daya

Prinsip dasar dari sebuah *boost converter* terdiri dari 2 kondisi yang berbeda yaitu. Bila saklar ditutup , arus mengalir melalui induktor searah jarum jam dan induktor menyimpan energi . Polaritas dari sisi kiri induktor adalah positif. Ketika saklar di *on*, saklar S ditutup, maka akan mengakibatkan terjadinya peningkatan arus induktor. Ketika saklar dibuka , arus akan berkurang sebagai impedansi lebih tinggi. Oleh karena itu, perubahan atau pengurangan saat ini akan ditahan oleh induktor. Dengan demikian polaritas akan terbalik (berarti sisi kiri

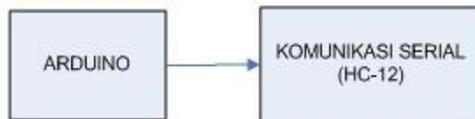
induktor akan negatif sekarang). Akibatnya dua sumber akan dalam seri menyebabkan tegangan yang lebih tinggi untuk mengisi kapasitor melalui dioda D, maka ketika sakelar *off*, saklar terbuka dan satu-satunya jalan yang ditawarkan untuk arus induktor adalah melalui *flyback* dioda D, kapasitor C dan beban R. Konfigurasi Arduino Nano dengan IC *boost converter* dapat dilihat pada Gambar 3.6



**Gambar 3.6** Boost Converter ke Arduino Nano

### 3.1.4 Pengkabelan Arduino Nano dengan HC-12

Perencanaan komunikasi data pada Tugas Akhir ini menggunakan HC-12, yang berfungsi untuk mengirimkan data dari *stick commander* ke *camera crane* dan *2 axis pan-tilt head*. Dimana data yang dikirimkan adalah data *joystick* dan data sensor MPU6050. Diagram blok dapat dilihat pada Gambar 3.7



**Gambar 3.7** Diagram Blok Komunikasi

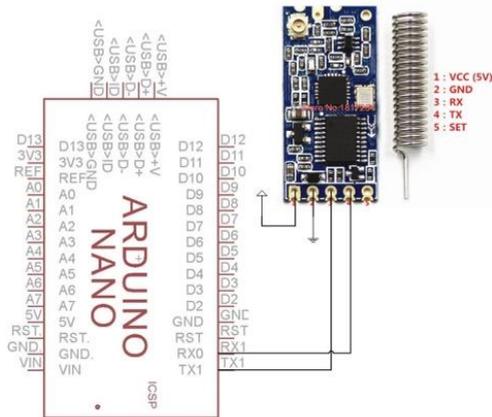
Terdapat 5 pin pada modul HC-12 yaitu VCC, GND, Rx, Tx, SET. Pengkabelan HC-12 dengan Arduino dapat dilihat pada gambar 3.5 Konfigurasi pin HC-12

No	Pin	Keterangan
1.	Vcc	Masukan catu daya, DC3.2V-5.5V, dengan beban kapasitas tidak kurang dari 200mA.
2.	GND	<i>Ground</i>
3.	RxD	Masukan data UART, tingkat TTL. 1k resistor dihubungkan secara seri di dalam modul
4.	TxD	UART data <i>output</i> , level TTL.

		Resistor 1k dihubungkan secara seri di dalam modul
5.	SET	Parameter <i>setting</i> kontrol pin, aktif <i>low level</i> . Resistor 1k dihubungkan secara seri di dalam modul
6.	ANT	433MHz antenna pin

Perintah AT digunakan untuk mengatur parameter modul dan beralih antar modul berfungsi saat modul berada dalam mode perintah. Setelah ditetapkan, perubahan ini akan menjadi *valid* hanya setelah keluar dari mode perintah. Parameter disimpan di memori *flash onboard non-volatile*, jadi tidak akan hilang saat *power* dilepas. Perintah AT digunakan untuk mengatur parameter modul dan beralih antar modul berfungsi saat modul berada dalam mode perintah. Setelah ditetapkan, perubahan ini akan menjadi *valid* hanya setelah keluar dari mode perintah. Parameter disimpan di memori *flash onboard non-volatile*, jadi tidak akan hilang saat *power* dilepas. Ada dua cara untuk masuk ke mode perintah:

- Saat terhubung dengan catu daya, tarik Pin 5 ("SET") rendah, tunggu 40ms untuk mode perintah untuk dilibatkan
- lepaskan catu daya, sambungkan Pin 5 ("SET") ke GND, kembali energi modul



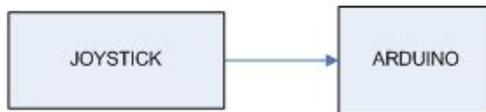
**Gambar 3.8** Konfigurasi HC-12 dengan Arduino Nano

### 3.1.5 Pengkabelan Arduino Nano dengan Modul Joystick

Perancangan Tugas Akhir ini menggunakan modul *joystick dual axis x y*, untuk menggerakkan motor servo pada *pan-tilt head*. Nilai analog dari joystick akan di ubah menjadi nilai digital oleh Arduino

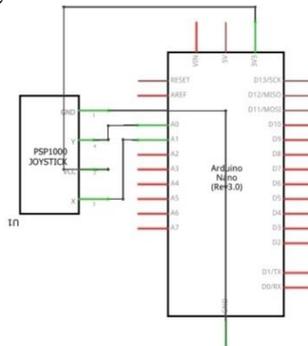
Modul *joystick* memiliki 5 pin yaitu GND, +5V, Vx, Vy dan SW. Untuk pin Vx, Vy di hubungan ke pin analog pada Arduino, dan SW ke pin digital pada Arduino. Modul *joystick* terdiri dari variabel resistor yang memiliki nilai resistansi yang akan diubah ke nilai ADC. Nilai ADC maksimal yang di terbaca pada serial monitor adalah 1024. Pengkabelan *joystick* dengan Arduino dapat dilihat pada gambar 3.5

Prinsip kerja modul *joystick* adalah terdapat dua buah *trimmer* (variabel resistor) dan sebuah *toggle switch*. Pada modul *joystick* memiliki trimmer dengan resistansi 10K, *joystick* memiliki 2 trimmer, satu untuk gerakan vertikal (atau maju-mundur) dan satu lagi untuk gerakan horisontal (atau kanan-kiri).



**Gambar 3.9** Diagram Blok Arduino dengan joystick

Pada Gambar 3.9 merupakan diagram blok joystick dengan Arduino. Pin A0 dan A1 pada Arduino digunakan untuk membaca nilai tegangan pada kedua *trimmer* yang berfungsi untuk pergerakan maju-mundur dan kanan-kiri. Dengan demikian, hasil yang didapatkan mirip dengan titik koordinat, yaitu koordinat X dan koordinat Y.



**Gambar 3.10** Konfigurasi Joystick dengan Arduino Nano

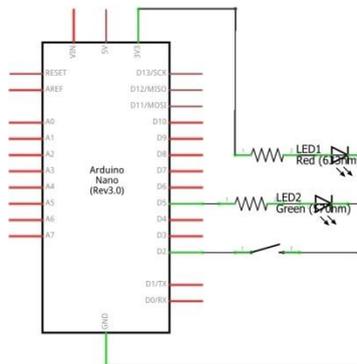
### 3.1.6 Perancangan Rangkaian Push Button dan LED

Perancangan Tugas Akhir ini menggunakan *push button* untuk mengaktifkan sensor *accelero* dan *gyro*. Cara kerja push button pada *stick commander* adalah mengaktifkan sensor *accelero* dan *gyro*, saat *push button* ditekan maka terdapat pembacaan data pada sensor dan mengirimkan. Namun saat *push button* dilepas, akan berhenti membaca data sensor dan mengirimkan data. Pada *camera crane* akan terbaca posisi terakhir saat *push button* ditekan.

Terdapat dua LED sebagai indikator, LED pertama berwarna merah sebagai indikator bahwa *stick commander* sedang aktif dan LED kedua berwarna hijau sebagai indikator sensor MPU6050 seang aktif hal itu juga saat push button diteka. Untuk pengkabelan LED dengan Arduino yaitu LED 1 (berwarna merah) sebagai indikator aktif tidaknya *stick commander* terhubung pada VCC dan GND. Untuk LED 2 (berwarna hijau) sebagai indikator telah aktif sensor MPU6050. VCC pada LED 2 terhubung pada D5 dan terhubung pada GND. Diagram blok led, push button dengan arduino dapat dilihat pada gambar 3.11 .Pengkabelan dapat dilihat pada Gambar 3.12



Gambar 3.11 Diagram Blok Led dan Push Button

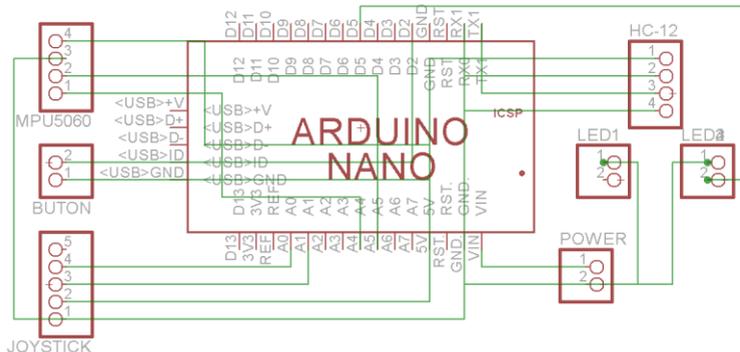


Gambar 3.12 Konfigurasi 2-Led Indikator dengan Arduino Nano

### 3.1.7 Perancangan Rangkaian *Shield* Arduino Nano

Pada Tugas Akhir ini menggunakan rangkaian *shield* untuk menghubungkan semua komponen dengan efektif. Pada perancangan rangkaian controller ini menggunakan Arduino Nano untuk menerima masukan dari sensor MPU6050, *joystick*, dan mengirimkan data ke *camera crane* melalui komunikasi nirkabel dengan menggunakan modul HC-12.

Pada rangkaian ini ditambahkan sebuah *shield* berupa rangkaian LED dengan push button. Pada *shield* ini terdapat port analog dan port digital yang dihubungkan dengan port pada Arduino. Arduino sebagai mikrokontroler diprogram agar dapat membaca nilai sensor accelero dan gyro, saat *push button* di tekan. Dan membaca nilai *joystick* untuk di kirimkan ke *camera crane* melalui modul HC-12. Tombol *switch* untuk meng aktifkan *stick commander* untuk dapat bekerja. Terdapat dua LED sebagai indikator saat pengambilan data dan data telah dikirim. Perancangan rangkaian *shield* Arduino Nano yang diperlukan oleh alat ini. Rangkaian *shield* Arduino Nano dengan sistem bisa di lihat pada Gambar 3.8



**Gambar 3.13** Rangkaian *Shield* Arduino Nano

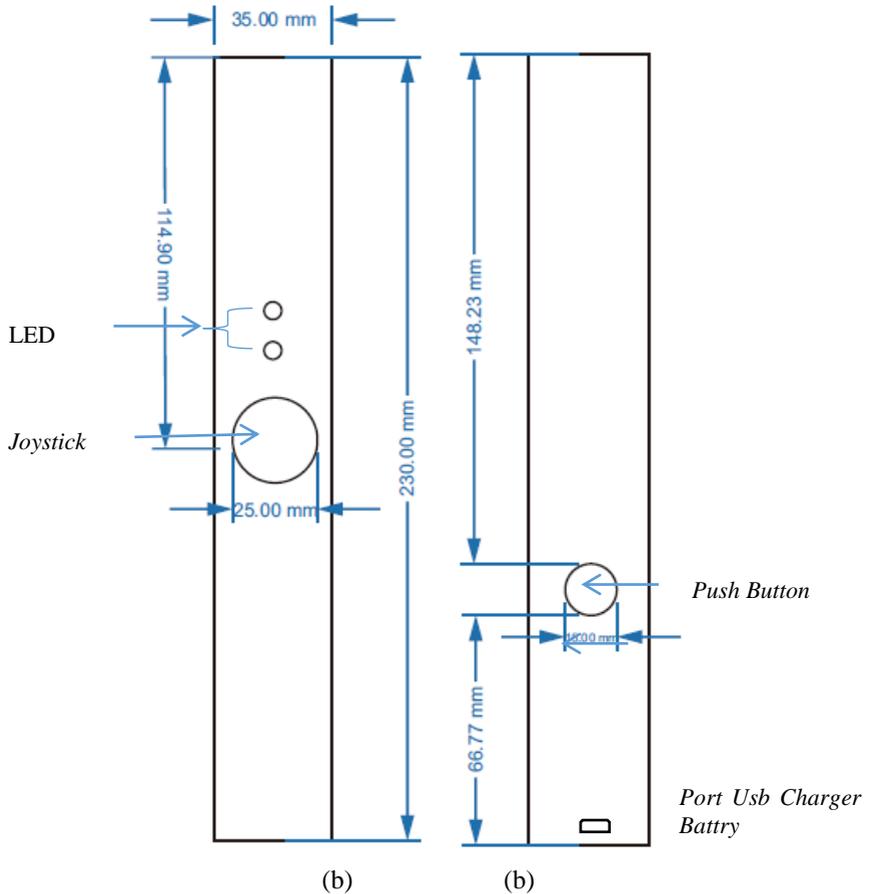
**Tabel 3.1** Mapping pin analog dan digital yang digunakan

No	Modul	Pin Analog/Digital
1.	MPU6050	VCC GND SCL = Pin Analog 5 SDA =Pin Analog 4

2.	Joystick	VCC = 5Volt GND Vrx = Analog 0 Vry = Analog 1
3.	HC-12	VCC =5V GND TX= Digital 0 (Rx) RX= Digital 1 (Tx)
.4.	Push Button	VCC= 5V (Digital 2) GND= GND
5.	LED	LED 1 VCC = VIN GND = GND LED 2 VCC = 5V(Digital 5) GND = GND

### 3.1.8 Perancangan Hardware (Perangkat Keras) *Stick Commander*

Perancangan *Stick Commander* dirancang sedemikian rupa agar dapat digunakan dengan satu tangan. Pada Gambar 3.9 merupakan desain dari *Stick Commander* yang tampak depan dan belakang. Pada bagian depan *Stick Commander* terdapat *joystick*, dan dua LED. Pada bagian belakang terdapat *push button* dan *port usb* modul *charger* baterai LiPO sebagai catu daya dari *Stick Commander*.



**Gambar 3.14** Desain *Stick Commander* tampak Depan (a) dan Belakang (b)

### 3.2 Perancangan *Software Stick Commander*

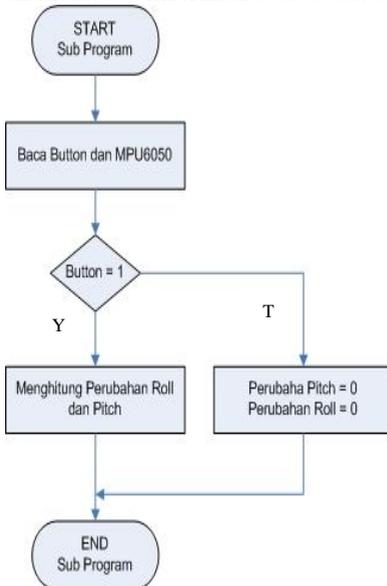
Dalam pembuatan perangkat lunak pada *stick commander* difungsikan untuk memprogram Arduino menggunakan *Arduino IDE*.

#### 3.2.1 *Flowchart*

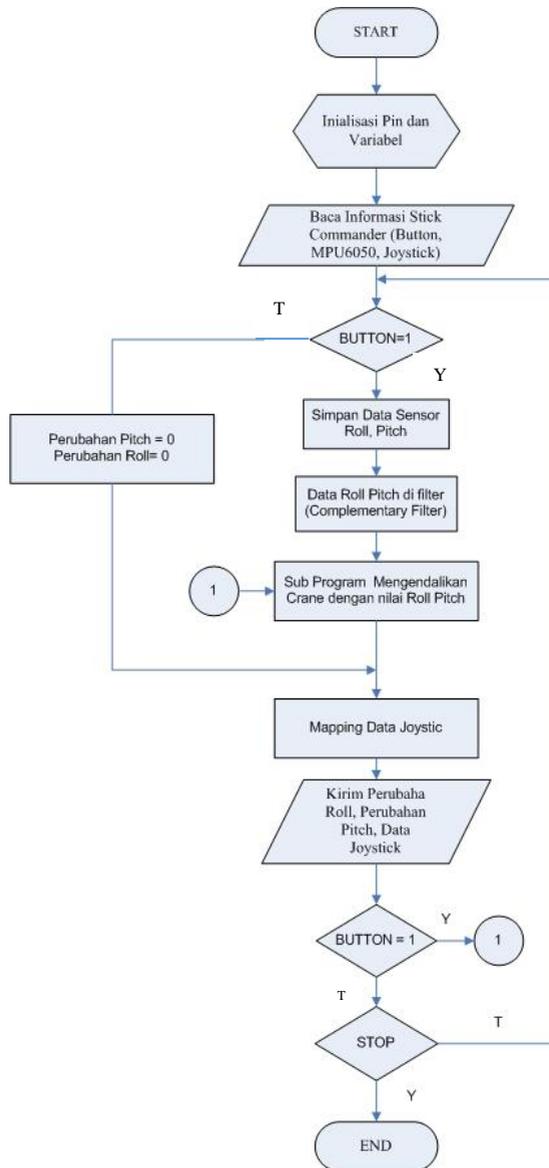
*Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* merupakan cara penyajian dari suatu algoritma. Bagan ini dinyatakan

dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. Gambar 3.16 merupakan gambaran *flowchart* dari program yang terdapat pada *mikrokontroler*.

Pada Tugas Akhir ini terdapat *flowchart* program dari *stick commander*. Dilakukan pendeklarasian variabel yang akan digunakan pada program. Membaca informasi pada *stick commander* yang berupa nilai sensor MPU6050, *joystick*, dan *push button*. Selanjutnya akan melalui kondisi yang menghasilkan beberapa kemungkinan. Ketika benar *button*=1 maka akan terjadi proses menyimpan data MPU6050. Kemudian masuk proses sub program untuk mengendalikan *camera crane* dengan menemukan nilai perubahan *roll* dan *pitch*, setelah itu masuk ke proses *mapping* data *joystick*. Jika *button* tidak sama dengan 1 maka nilai dari perubahan *roll* dan *pitch* adalah 0, masuk ke proses *mapping* data *joystick*. Kirim data *joystick*, perubahan *roll* dan perubahan *pitch*. Setelah itu jika kondisi *button* masih sama dengan 1 maka akan dikembalikan ke sub program. Jika tidak *stop* program akan kembali membaca informasi *stick commander*.



**Gambar 3.15** *Sub Program*



**Gambar 3.16** Flowchart Program *Stick Commander*

### 3.2.2 Pendeklarasian Variabel

Perancangan perangkat lunak Tugas Akhir, yang pertama dilakukan adalah pendeklarasian variable untuk mempermudah menginisialisasi program, bisa dilihat pada Gambar 3.12. Pustaka yang digunakan adalah `wire.h` yang merupakan pustaka dari I2C, berfungsi untuk menangani protokol serial sinkron secara I2C yang telah disediakan oleh Arduino IDE. Kemudian akan pendeklarasian variabel dan pin. `xPin` terhubung dengan pin analog 1 atau A1, untuk dapat membaca nilai analog dan `yPin` terhubung dengan pin analog 0 atau A0. Dan beberapa variable lain yang di deklarasikan. `const int MPU_addr=0x68` merupakan pendeklarasian alamat register MPU. Alamat default dari MPU-6050 adalah 0x68, alamat ini dijadikan alamat untuk MPU-6050 pertama sebagai pemberi nilai *set point*.

```
#include<Wire.h>

int xPin = A1;
int yPin = A0;
int a = 0;
int b = 0;
const int MPU_addr=0x68;
double AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
uint32_t timer;
double compAngleX, compAngleY;
int dataAwalX = 0;
int dataAwalY = 0;
int rangeX= 0;
int rangeY= 0;
#define degconvert 57.2957786
```

**Gambar 3.17** Deklarasi Variable

### 3.2.3 Perancangan Software Joystick

Perancangan perangkat lunak Tugas Akhir menggunakan program *joystick*. Pada Gambar 3.18 merupakan ini merupakan program dari *joystick* yang berfungsi untuk mengendalikan motor servo pada 2 *axis pan-tilt head*. Tipe data *integer* (bilangan bulat) `xp`, terdapat `analogRead`

berfungsi untuk membaca nilai analog pada input pin yang akan menghasilkan nilai integer antara 0-1023 pada program pembacaan nilai analog pada xPin. Kemudian, terdapat *mapping* digunakan untuk memetakan suatu nilai dari *range* tertentu ke *range* yang lain. Nilai dari xp merupakan hasil dari pemetaan nilai adc 0 sampai 1023 dari potensio pada sumbu Y, menjadi nilai -10 sampai 10. Fungsi tersebut juga digunakan pada *joystick* sumbu Y. Potongan program untuk *joystick* dapat dilihat pada gambar 3.13

```
void loop()
{
  filter();
  int xp = analogRead(xPin);
  xp = map(xp, 0, 1023, -10, 10);
  int yp = analogRead(yPin);
  yp = map(yp, 0, 1023, -10, 10);
```

**Gambar 3.18** Program Fungsi *Joystick*

### 3.2.4 Perancangan Software Komunikasi Menggunakan HC-12

Perancangan perangkat lunak komunikasi pada Tugas Akhir ini menggunakan HC-12. Sebelum menggunakan HC-12 harus dilakukan perintah pengaturan modul dengan perintah *AT Command*. Mencakup perintah AT, beberapa perintah AT yang digunakan adalah AT-DEFAULT yang memiliki pengaturan seperti di bawah ini:

- AT-BR9600, mengatur nilai *baud rate* menjadi 9600
- AT-C001, perintah ini untuk mengatur kanal mana yang akan digunakan
- AT-FU3, Untuk mengubah modus transmisi *port serial* dari modul. FU3 memiliki arus 16mA, *delay* transmisi 4-80ms dan pengoprasian pada daya maksimal 600m pada 9600bps.
- AT+P8 merupakan pengaturan untuk nilai dari daya pemancar modul yaitu 20Dbm (100mW).

Setelah mengatur perintah *AT Command* pada modul HC-12, selanjutnya adalah program komunikasi pada Arduino IDE. Berikut merupakan program komunikasi. Program komunikasi menggunakan UART menggunakan pin Arduino Tx terhubung pada pin Rx modul

HC-12 dan Rx Arduino terhubung pada pin Tx HC-12. Sehingga untuk mengirimkan data hanya menggunakan perintah `Serial.print`. Data yang dikirim berupa nilai `rangeY`, `a`, `b`, `rangeX`, `xp`, dan `yp`.

```
Serial.print("*");
Serial.print(rangeY);
Serial.print(",");
Serial.print(a);
Serial.print(",");
Serial.print(b);
Serial.print(",");
Serial.print(rangeX);
Serial.print(",");
Serial.print(xp);
Serial.print(",");
Serial.print(yp);
Serial.println("#");
```

**Gambar 3.19** Program *Transmitter* Komunikasi Serial

### 3.2.5 Perancangan Software Push Button

Perancangan perangkat lunak Tugas Akhir ini menggunakan beberapa program untuk perintah *push button*. *Push button* pada *stick commader* berfungsi untuk mengaktifkan sensor *accelero* dan *gyroscope*. Saat *push button* ditekan maka nilai sudut akan dikirimkan ke *camera crane*. Perintah program yang digunakan agar *push button* menggunakan perintah *interupt*

```
pinMode(2, INPUT_PULLUP);
attachInterrupt(0, interruptnol, FALLING);
delay(100);
```

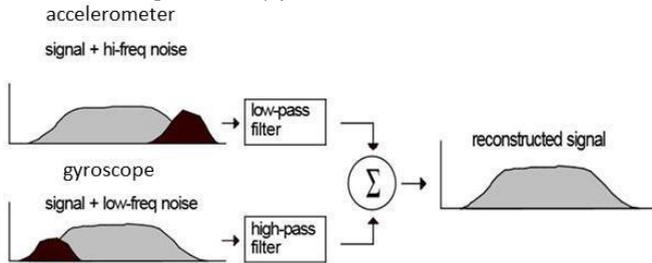
**Gambar 3.20** Inialisasi Pin *Interupt*

*Interupt* Merupakan intruksi untuk mengaktifkan interupsi dari luar, pin yang berhubungan dengan interupsi dari luar pada Arduino. Perintah berfungsi berperan sebagai fungsi penghandel interupsi dan mode berupa *low*, *change*, *rising*, dan *falling*. Interupsi eksternal dapat dibangkitkan apabila ada perubahan logika baik transisi naik (*rising edge*) maupun transisi turun (*falling edge*) pada pin interupsi.

### 3.2.6 Perancangan Software Complementary filter

Perancang perangkat lunak Tugas Akhir ini menggunakan beberapa program untuk *complementary filter*. Complementary filter adalah filter yang dirancang untuk menggabungkan dua sensor *accelero* dan *gyroskop*. Pada *stick commander* di gunakan sensor *accelero* dan *gyro* untuk membaca kemiringan dari *stick commander*. Sensor accelerometer dapat mengukur perubahan sudut berdasarkan nilai percepatan grafitasi tetapi dipengaruhi oleh vibrasi dan akselerasi. Sensor gyroscope dapat mengukur sudut dari proses integral kecepatan sudut tetapi dipengaruhi oleh sinyal drift yang tidak beresilasi. Oleh karena itu metode untuk melakukan kombinasi kedua sensor tersebut agar dapat mengukur dengan lebih akurat. Sifat vibrasi dan akselerasi sensor *accelerometer* adalah mempunyai frekuensi tinggi, sedangkan sinyal *drift* pada *gyroscope* mempunyai frekuensi yang rendah.

*Complementary filter* dapat digunakan untuk kombinasi data *accelero meter* dan data *gyroscope* terintegrasi melewati *low-pass* dan yang terakhir melalui *high-pass*, pada Gambar 3.21 merupakan ilustrasi sinyal derau pada sensor *accelerometer* dan *gyroscope*. Metode yang digunakan adalah *Complementary filter*.



**Gambar 3.21** Blok Diagram Proses Complementary Filter

Jika *complementary filter* ini diaplikasikan ke bahasa pemrograman maka menggunakan rumus di bawah ini :

$$sudut = (a) \times (sudut + gyro \times dt) + (a - 1) \times (acc) \dots\dots\dots(3.1)$$

Dimana nilai

- a : koefisien filter
- dt : waktu sampling, disesuaikan dengan waktu sampling sensor
- Sudut : sudut keluaran complementary filter
- gyro : keluaran sensor *gyroscope* berupa kecepatan sudut
- acc : keluaran sensor accelerometer yang sudah berupa sudut

Berikut merupakan bagian program dari *Complementary Filter*. Pada Tugas Akhir ini menggunakan 2 axis yaitu X(*roll*) dan Y(*pitch*). Untuk memperoleh nilai *roll* dan *pitch* menggunakan fungsi Arc Tangen atau *Atan2* merupakan fungsi yang menghasilkan nilai Arc Tangen dalam empat kuadran dari satu koordinat X dan Y tertentu atau diantara dua koordinat. *Atan2* menghasilkan nilai sudut(dalam suatu radian) antara baris  $y=0$  dan garis yang dihubungkan dengan koordinat tertentu. Fungsi dalam program *roll* merupakan nilai *atan2* dari *accelero Y* dan *accelero Z* dikalikan *degconvert* agar hasilnya dari radian menjadi derajat.

Pada listing program selanjutnya untuk menemukan nilai dari *gyroscope*. *GyX* nilai keluaran dari sensor *gyroscope* pada sumbu X dibagi dengan sensitifitas nilai ADC sensor yaitu 131. *Listing program* selanjutnya adalah program *complementary filter* yang merupakan program penggabungan dari nilai *gyroscope* dan *accelerometer*, dengan rumus fungsi pada . Pada Tugas Akhir ini menggunakan koefisien filter 0,99.

```
double dt = (double)(micros() - timer) / 1000000;
timer = micros();
double roll = atan2(AcY, AcZ)*degconvert;
double pitch = atan2(-AcX, AcZ)*degconvert;
double gyroXrate = GyX/131.0;
double gyroYrate = GyY/131.0;
compAngleX = 0.99 * (compAngleX + gyroXrate * dt) + 0.01 * roll;
compAngleY = 0.99 * (compAngleY + gyroYrate * dt) + 0.01 * pitch;
```

**Gambar 3.22** Program *Complementary Filter*

## **BAB IV**

### **HASIL PERANCANGAN DAN PENGUJIAN**

Pada bab ini akan membahas mengenai hasil pengujian dan analisa atas penyusunan alat untuk “Perancangan *Digital Stick Commander* untuk pengendali camera crane berbasis *Accelero* dan *Gyro Sensor*”. Data pengujian sangat diperlukan untuk implementasi dalam dunia nyata. Kinerja suatu sistem sangat dipengaruhi oleh kinerja per bagian dari sistem tersebut. Pengujian merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil-hasil yang dicapai pada 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 atau kesalahan dalam sistem.

Pengujian tersebut meliputi :

- a. Pengujian *Input/Output* Arduino
- b. Pengujian *Boost Converter*
- c. Pengujian sensor IMU tipe MPU6050
- d. Pengujian modul *joystick*
- e. Pengujian Modul HC-12
- f. Pengujian *Complementary Filter*
- g. Pengujian *Stick Commander* terhadap *Camera Crane* dan *Pan Tilt Head*

#### **4.1 Pengujian Input/Output Arduino**

Pengujian Mikrokontroler Arduino Nano bertujuan untuk mengetahui tegangan output pada saat diberi *Input High Voltage* dan *Input Low Voltage* pada setiap 14 pin digital Arduino Nano.

Cara pengujian ini dilakukan pada waktu yang bersamaan ketika Vcc dari mikrokontroler Arduino Nano mendapat Input tegangan sebesar 5 volt dari Power Supply. Hasil dari pengukuran tegangan tersebut terdapat pada tabel 4.1

**Tabel 4.1** Hasil Pengujian *Input Output* Arduino Nano

	High (volt)	Low (volt)
1.	4,98	0.00
2.	4,98	0.00
3.	4,98	0.00
4.	4,98	0.00
5.	4,98	0.00
6.	4,98	0.00
7.	4,98	0.00
8.	4,98	0.00
9.	4,98	0.00
10.	4,98	0.00
11.	4,98	0.00
12.	4,98	0.00
13.	4,98	0.00

#### 4.2 Pengujian Power Supply dan *Boost Converter*

Pengujian *Boost Converter* bertujuan untuk mengetahui tegangan *output* dari *Boost Converter* yang akan menjadi tegangan *input/Supply* Arduino Nano .

Cara pengujian *Boost Converter* dengan mengukur tegangan *output* baterai atau tegangan input *Boost Converter*. Setelah melalui rangkaian *Boost Converter* saat tanpa beban dan mengukur tegangan *output* setelah dibebani Arduino

**Tabel 4.2** Hasil Pengujian *Boost Converter*

Tegangan Input (Volt)	Tegangan Output (Volt)	Duty Cycle (%)
3.82	4,54	16
	5	24
	5,55	31
	6.15	38
	6,47	41
	7.32	48
	7,53	49
	8.17	53

$$\%Error = \frac{[Tegangan\ Keluaran - Tegangan]}{Tegangan} \times 100\% \dots\dots\dots (4.1)$$

**Tabel 4.3** Hasil Pengujian *Power Supply*

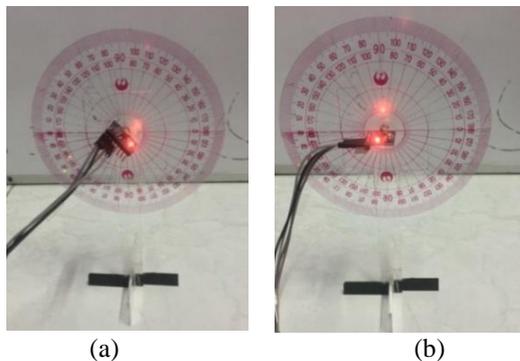
Tegangan	Tegangan Keluaran		Kesalahan (%)	
	Tanpa beban	Berbegan	Tanpa beban	Berbegan
+5 Volt	5.01 Volt	4,94 Volt	0,2 %	0,2%
+9 Volt	8.96 Volt	8.94 Volt	0,7 %	0,4 %

Hasil dari pengujian yang dilakukan menunjukkan bahwa *duty cycle* minimal 16% tegangan 4.54 Volt. Tegangan 5Volt pada *duty cycle* 24% yang digunakan dalam sistem . Perhitungan *duty cycle* dapat dilihat pada Tabel 4.2

Tegangan baterai yang telah melalui *boost converter* sebagai tegangan *supply*. *Power Supply* dapat digunakan sebagai sumber untuk mensuplai peralatan tersebut yang digunakan.

### 4.3 Pengujian Sensor IMU Tipe MPU6050

Pengujian sensor IMU Tipe MPU6050, bertujuan untuk mengetahui nilai dari sudut kemiringan sensor. Pengujian dilakukan dengan cara menghubungkan MPU6050 ke Arduino Nano dengan perintah program dan data akan tampil pada serial monitor. Pengujian terhadap sumbu Y(*pitch*)-X(*roll*), dilakukan dari sudut 0° hingga 180°. Untuk melihat hasil dilihat di *serial monitor*.



**Gambar 4.1** Cara Pengujian MPU6050 Sumbu-X (a) dan Sumbu Y (b)

**Tabel 4.4** Hasil P engujian MPU6050 pada *Roll*,

Sudut Busur	Pengukuran Sudut Sensor ( <i>Roll</i> )				Selisih Nilai Sudut
	Data 1	Data 2	Data 3	Rata rata nilai <i>roll</i>	
0	1.94	1.29	0.78	1.33	1.33
10	10.54	10.67	10.77	10.66	0.66
20	20.67	20.21	20.66	20.51	0.51
30	30.36	31.11	31.56	31.01	1.01
40	39.95	39.88	40.20	40.01	0.01
50	47.02	49.89	50.12	49.01	0.99
60	59.39	58.77	60.50	59.55	0.45
70	69.63	68.57	70.21	69.47	0.53
80	80.17	79.83	79.33	79.77	0.23
90	87.76	89.39	89.78	88.97	1.03
100	99.81	100.52	100.21	100.18	0.18
110	108.31	111.30	109.15	109.58	0.42
120	119.20	120.97	121.32	120.49	0.49
130	131.79	130.16	131.93	131.29	1.29
140	141.51	139.19	140.59	140.43	0.43
150	151.53	150.82	149.97	150.77	0.77
160	160.52	160.25	158.94	159.90	0.1
170	170.33	169.12	168.98	169.47	0.53
180	179.14	178.98	179.54	179.22	0.78

**Tabel 4.5** Pengujian MPU6050 pada *Pitch*

Sudut Busur	Pengukuran Sudut Sensor ( <i>pitch</i> )				Selisih Nilai Sudut
	Data 1	Data 2	Data 3	Rata Rata	
0	0.89	1.89	2.87	1.88	0.88
10	10.92	10.77	11.61	11.1	1.1
20	20.27	18.98	19.56	19.60	0.4
30	30.21	29.16	29.76	29.71	0.29
40	40.48	39.56	39.89	39.97	0.03
50	48.89	49.89	50.45	49.74	0.26
60	59.32	59.78	60.35	59.81	0.19
70	70.45	69.77	68.97	69.73	0.27
80	80.34	76.60	79.98	78.97	1.03

Sudut busur	Data 1	Data 2	Data 3	Rata Rata	Selisih nilai sudut
90	90.23	86.43	89.83	88.83	1.17
100	101.23	96.30	99.68	99.07	0.93
110	110.76	109.81	108.31	109.62	0.38
120	120.02	119.73	118.67	119.47	0.53
130	130.23	127.08	129.78	129.03	0.97
140	140.24	139.59	141.08	140.30	0.3
150	149.15	150.11	152.23	150.49	0.49
160	158.67	159.90	160.23	159.6	0.4
170	170.48	169.35	169.73	169.85	0.15
180	179.34	177.98	179.34	180.8	0.8

Hasil pengujian nilai sudut dari sensor *accelero* dan *gyro* terdapat dua yaitu *roll* dan *pitch*. *Roll* pergerakan pada sumbu X dan *Pitch* merupakan pergerakan pada sumbu Y. Pada data hasil pengujian diatas adalah konsisi saat sensor tidak melalui kombinasi nilai *accelero* dan *gyro* atau *complementary filter*. Pengujian dilakukan dari sudut 0° hingga 180°, dilakukan tiga kali pengujian.

Pada gerakan memutar ke kanan atau *roll* didapat data perubahan sudut tiga kali percobaan yang masing masing perubahan sudut di dapatkan nilai rata-rata pada sudut sensor. Selisih nilai sudut sebenarnya dan yang terbaca 0° sampai 2°. Begitu pada gerakan *pitch* memiliki perubahan nilai sudut 0° sampai 2°.

#### 4.4 Pengujian Modul Joystick

Pengujian Modul Joystick bertujuan untuk mengetahui nilai ADC dari modul *joystick*, serta mengetahui respon *pan tilt head* terhadap modul joystick

Pertama pengujian modul *joystick* dilakukan dengan cara menggerakkan joystick (atas, bawah, kanan dan kiri) kemudian diukur tegangan joystick dan nilai ADC yang dapat dilihat pada serial monitor setiap pergerakannya. Pengujian respon *joystick* pada *pan tilt head* dengan cara menghitung waktu pergerakan dari keadaan normal hingga bergerak 90 derajat.

**Tabel 4.6** Pengujian Joystick

	Nilai ADC		Tegangan (Volt)	
	Vx	Vy	Vx	Vy
Normal	0	0	2.46	2.56
Atas Maksimal	10	0	1.5mV	2.56
BawahMaksimal	-10	0	5	2.56
KananMaksimal	0	10	2.46	1.3Mv
Kiri Makasimal	0	-10	2.46	5

**Tabel 4.7** Hasil Pengujian Respon *Joystick*

Joystick	Pan	Tilt	Waktu Respon
Kanan	Bergerak	Tidak Bergerak	9.04 Sekon
Kiri	Bergerak	Tidak Bergerak	9.70 Sekon
Normal	Tidak Bergerak	Tidak Bergerak	-
Atas	Tidak Bergerak	Bergerak	10.58 sekon
Bawah	Tidak Bergerak	Bergerak	10.27 sekon

Dari hasil pengujian di atas menunjukkan tegangan maksimal yang masuk ke *joystick* adalah 5 Volt. Saat dalam keadaan normal(ditengah) tegangan yang masuk terbagi menjadi dua antara Vx 2.46 Volt dan pada Vy 2.56 Volt. Ketika nilai dari ADC adalah 0 pada Vx maka nilai tegangan 2.46 Volt, sedangkan pada Vy nilai nya 2.56 Volt. Selain itu jika nilai ADCnya -10 nilai pada Vx dan Vy adalah 5 Volt dan saat ADC bernilai 10 tegangan yang masuk kecil berupa milivolt. Hasil pengujian respon *joystick* terhadap *pan tilt head* memiliki rata-rata waktu respon 9.85 sekon

#### 4.5 Pengujian Modul HC-12

Pengujian Modul HC-12 bertujuan untuk mengetahui seberapa jauh komunikasi HC-12 ini bisa mengirim data dan menerima data dengan baik.

Pengujian HC-12 ini dengan cara pengujian modul HC-12 cara melihat koneksi antara HC-12. Ada beberapa tahapan dalam pengujian HC-12 yaitu pengujian *channel* yang berbeda, pengujian *channel* yang sama, pengujian pada *baud rate* berbeda dengan jarak tertentu tidak ada halangan, pengujian pada satu *baud rate dengan* jarak yang berbeda, dan pengujian dengan halangan

**Tabel 4.8** Pengujian *channel* berbeda

No	Channel Transmitter	Channel Reciver	Kondisi
1.	001	005	Tidak Terkirim
2.	001	002	Tidak Terkirim

**Tabel 4.9** Pengujian *channel* sama

No	Channel Transmitter	Channel Reciver	Kondisi
1.	100	100	Terkirim
2.	010	010	Terkirim

**Tabel 4.10** Pengujian dengan 3 *Baud Rate* Berbeda

Jarak (meter)	Kondisi (Baud Rate)					
	9600		19200		38400	
	Kondisi	Loss Data	Kondisi	Loss Data	Kondisi	Loss Data
5	Terkirim	Tidak	Terkirim	Tidak	Terkirim	Tidak
10	Terkirim	Tidak	Terkirim	Tidak	Terkirim	Tidak
15	Terkirim	Tidak	Terkirim	Tidak	Terkirim	Tidak
20	Terkirim	Tidak	Terkirim	Ada	Terkirim	Ada
25	Terkirim	Tidak	Terkirim	Ada	Terkirim	Ada
30	Terkirim	Tidak	Terkirim	Ada	Terkirim	Ada
35	Terkirim	Tidak	Terkirim	Ada	Terkirim	Ada

**Tabel 4.11** Pengujian pada *Baud Rate* 9600 dengan Delay 0.1s

Jarak (meter)	Kondisi	Loss Data
5	Terkirim	Tidak
10	Terkirim	Tidak
15	Terkirim	Tidak
20	Terkirim	Tidak
25	Terkirim	Tidak
30	Terkirim	Tidak
35	Terkirim	Tidak

**Tabel 4.12** Pengujian Terdapat Halangan dengan *Baud Rate* dengan Delay 0.1 s

Halangan	Kondisi	Loss Data
1 Dinding	Terkirim	Tidak
2 Dinding	Terkirim	Tidak
3 Dinding	Terkirim	Tidak
4 Dinding	Terkirim	Ada

Pengujian tahap pertama yaitu pengujian *channel* berbeda dan *channel* yang sama. Hasil pengujian dapat dilihat pada tabel 4.8 dan Tabel 4.7 menunjukkan bahwa, saat HC-12 yang akan dihubungkan memiliki *channel* yang berbeda tidak ada yang terhubung sekalipun *channel* yang digunakan berdekatan seperti contoh *channel* 001 dan 002. Pada saat *channel* di samakan antara *transmitter* dan *receiver* dapat tehubung dan dapat mengirimkan data

Pengujian tahap kedua yaitu pengujian dengan tiga macam *baud rate* yang berbeda beda 9600, 19200 dan 38400. Hasil pengujian dapat dilihat pada Tabel 4.10. *Baud rate* merupakan jumlah pulsa yang dikirimkan dalam setiap detik melalui saluran transmisi tertentu. Pada hasil pengujian dengan jarak mulai 5-35 meter dengan range 5 meter. Pada *baud rate* 9600 saat mengirim data semua terkirim dari jarak minimal hingga jarak maksimal yang ditentukan. Pada *baud rate* ini tidak terdapat *loss data*/data yang hilang. Pada *baud rate* 19800 saat mengirim data dari jarak 5-15meter pengiriman berjalan dengan lancar lebih cepat dari *baudrate* sebelumnya namun terdapat *loss data* saat jarak 20 – 35 meter. Pada *baud rate* 38400 jarak yang ditempuh 5-15 meter dan setelah itu jarak 20-35 meter terdapat *loss data*.

Pengujian tahap kedua menjadi acuan, pengujian tahap ketiga yang digunakan adalah *baud rate* 9600 dengan *delay* 0.1sekon. Pada pengujian ini jarak 5-35 meter *range data* 5 meter. Pada jarak 5-35 Meter data terkirim tidak terdapat *loss data*.

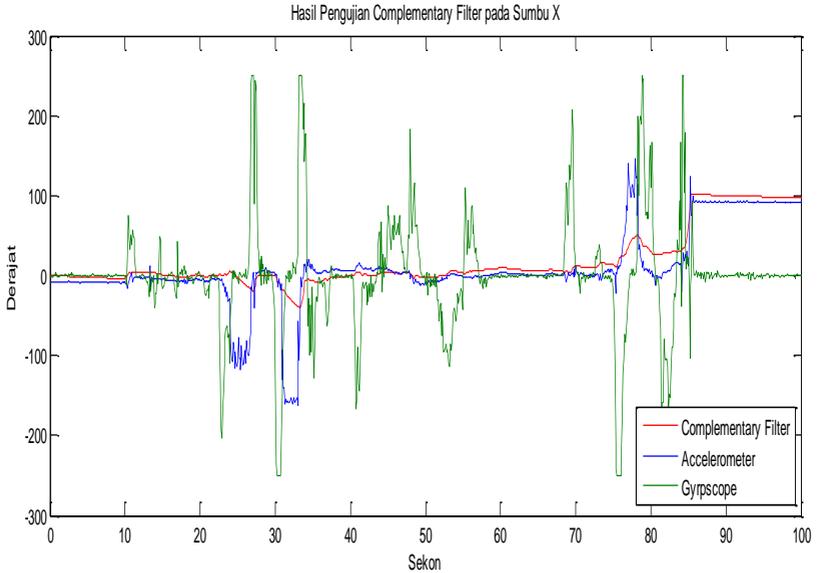
Pengujian tahap keempat adalah dengan pengiriman data terhadap halangan. Halangan yang digunakan adalah halangan dinding sampai 4 dinding. Pada 1-3 dinding tidak terdapat *loss data*

#### 4.6 Pengujian Complementary Filter

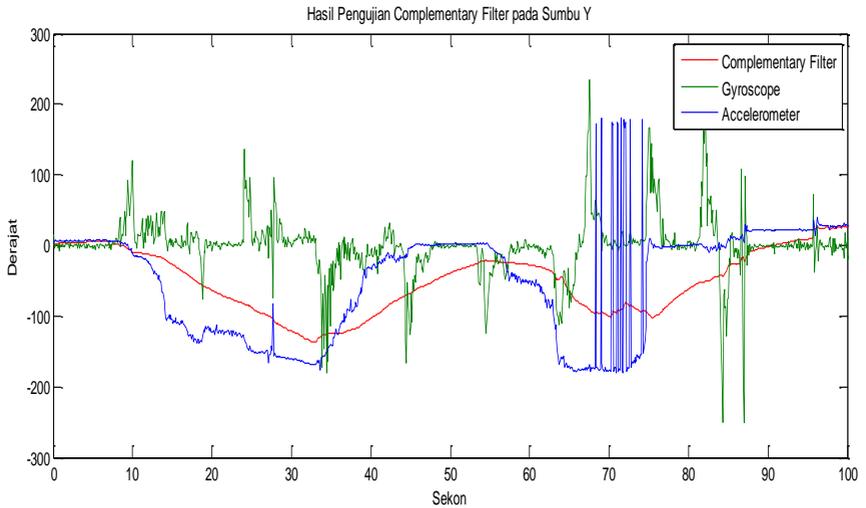
Pengujian *complementary filter* bertujuan untuk mengetahui keluaran dari sensor *accelero* dan *gyro*, serta membuktikan bahwa pada *stick commander* di butuhkan *Complementary Filter*.

Pengujian dengan menggunakan program arduino yang terdapat program *complementary filter* dan program *datalogger* microSD. Cara

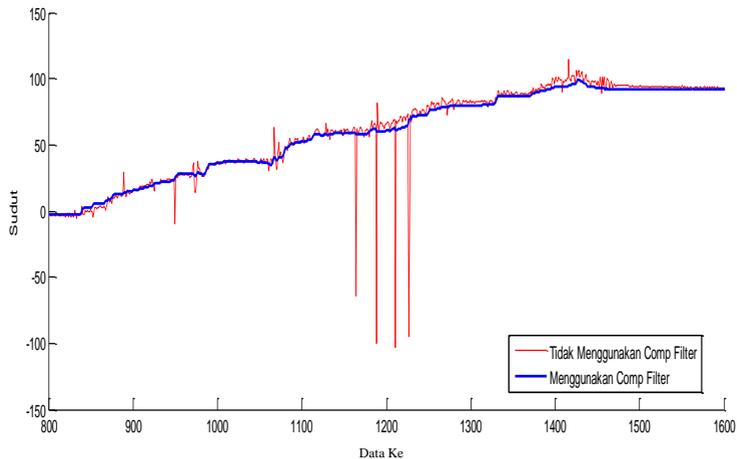
pengujian adalah saat diam dan saat digunakan. Saat diam dengan meletakkan *Stick Commander* di atas meja, kemudian diberikan getaran. Saat digunakan diberikan sedikit guncangan tangan seperti tangan orang yang tremor. Pengujian saat bergerak dengan menggunakan nilai dari sumbu Y dan pengujian saat diam menggunakan nilai dari sumbu X.



**Gambar 4.2** Grafik pengujian complementary filter saat diam pada sb X



**Gambar 4.3** Grafik pengujian complementary filter saat bergerak pada sb Y



**Gambar 4.4** Grafik Pengujian Complementary Filter

Dari pengujian pada pergerakan saat diam dan bergerak pada sumbu X (*Roll*). Dapat dilihat ada Gambar 4.2 pengujian saat diam

dimana masih terdapat nilai yang jauh berbeda ketika diberi getaran sensor MPU 6050 belum melalui. Sedangkan pada Gambar 4.3 merupakan hasil pengujian pergerakan pada sumbu Y (*pitch*). Saat *stick commander* diberi getaran maka keluaran dari nilai sudut masih beresilasi. Namun pada keluaran yang telah digabungkan tidak beresilasi begitu jauh. Hal tersebut juga terjadi pada pengujian pergerakan pada sumbu *roll*. Pada *stick commander* digunakan nilai dari accelerometer dan gyroscope yang telah bergabung untuk dikirimkan sebagai input penggerak *camera crane* dan *pan-tilt head*. Accelerometer tidak bisa mengikuti pergerakan yang cepat karena responnya lambat. Accelerometer dapat mengukur perubahan sudut berdasarkan nilai percepatan gravitasi tetapi dipengaruhi oleh vibrasi dan akselerasi.

#### 4.7 Pengujian *Stick Commander* terhadap *Camera Crane* dan *Pan Tilt Head*

Pengujian ini bertujuan untuk mengetahui, *stick commander* dapat bekerja dengan baik atau tidak. Dari respon pada pergerakan *Roll crane*. Pada *stick commander* terdapat sensor *accelerogyro* untuk menggerakkan *camera crane*. Dan menggerakkan *pan tilt head* menggunakan *joystick*.

Cara pengujian adalah dengan mengoperasikan *camera crane* dengan *stick commander* dengan jarak tertentu. Pengujian waktu respon pergerakan *roll* pada *camera crane* dengan jarak tertentu.

**Tabel 4.13** Hasil Pengujian *camera crane* dan *pan tilt* terhadap jarak

Jarak (Meter)	Pan	Tilt	Crane Pitch	Crane Roll
5	Bergerak	Bergerak	Bergerak	Bergerak
10	Bergerak	Bergerak	Bergerak	Bergerak
15	Bergerak	Bergerak	Bergerak	Bergerak
20	Bergerak	Bergerak	Bergerak	Bergerak

**Tabel 4.14** Hasil pengujian Respon pergerakan *roll* terhadap jarak

Jarak (Meter)	Crane	
	Respon Pitch (Bawah-Atas)	Respon Pitch (Atas-Bawah)
5	10.68 Sekon	10.24 Sekon
10	9.48 Sekon	10.10 Sekon
15	10.77 Sekon	10.83 Sekon
20	8.58 Sekon	8.12 Sekon

Pada hasil pengujian pada Tabel 4.13 menunjukkan saat *stick commander* digunakan pada jarak 5-20 masih dapat bekerja dengan respon yang baik. Pergerakan *pan* dan *tilt* bergerak dengan respon sesuai dengan hasil pengujian pengujian pada tabel 4.7. Pada hasil pengujian respon pergerakan *roll* terdapat dua respon dari bawah ke atas dan dari atas ke bawah. Pada pengujian *respon pitch* dari bawah ke atas memiliki rata rata 9.87 sekon, dari atas ke bawah memiliki rata rata 9.87 sekon.

## **BAB V PENUTUP**

Dari hasil penelitian kali ini dapat diambil kesimpulan antara lain sebagai berikut :

1. Pengiriman data dengan komunikasi serial HC-12 berjalan dengan baik pada jarak hingga jarak 35 meter pada kondisi tanpa penghalang dengan *delay* menggunakan *baud rate 9600*. Data yang dikirimkan, semua terkirim tanpa adanya *loss data*. Jika menggunakan halangan berupa 4 dinding saat pada berpembatas dinding ketiga terdapat *loss data*
2. Pada pengujian sensor MPU6050 memiliki nilai selisih sudut 0-2 derajat dari sudut terukur dan sebenarnya. Data sensor MPU6050 terdapat osilasi ketika diberi getaran. Namun keluaran sudut yang sudah melalui *complementary filter* memiliki keluaran nilai sudut yang tidak beresilasi.
3. *Mapping* data nilai ADC joystick yaitu -10 sampai 10. Respon *joystick* terhadap pergerakan *pan* dan *tilt* memiliki rata 9.8 sekon

Saran untuk penelitian selanjutnya yaitu *stick commander* didesain lebih mudah di gengam di tangan. Dan terdapat HMI dan Monitor menampilkan gambar yang diambil untuk mempermudah mengendalikan *camera crane* di ruangan yang berbeda.

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

## DAFTAR PUSTAKA

- [1] Ave Borregas. 2013, MPU-6000 and MPU-6050 Product Specification Revision 3.4, Inven Sense. Datasheet
- [2] Ave Borregas. 2013, MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2, Inven Sense. Datasheet
- [3] Atronics. 2014. Joystick Module. Future Eletronics. Datasheet
- [4] Colton, Shane,2015 The Balance Filter.: *A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform,,pdf*,
- [5] Fikri, Berlian. 2017. Perancangan dan Implementasi Real Segway pada Skateboard Roda Satu Menggunakan Gyroscope dan Accelerometer
- [6] Gunawan. 2009. Rancang bangun DC-DC Buck Converter, Sripsi, Universitas Indonesia
- [7] Jurnal Binus. Bab II Landasan Teori. [thesis.binus.ac.id/doc/Bab2Doc/2012-1-00332-SK%20Bab2001.doc](http://thesis.binus.ac.id/doc/Bab2Doc/2012-1-00332-SK%20Bab2001.doc). Diakses pada 21 Mei 2017
- [8] Rozee, Robert. 2016, HC-12 Wireless Serial Port Communication Module, User Manual version 2.3B
- [9] Siswoyo, Bambang, 2012. Belajar Arduino: pengantar, <http://bsiswoyo.lecture.ub.ac.id/2012/06/belajar-Arduino-pengantar/>, 02 Mei 2017
- [10] Widada, Wahyu.2016, Metode Adaptif Cut-Off untuk Complementary Filter Accelerometer dan Gyroscope untuk Sudut Pitch dan Roll Wahana Terbang.pdf
- [11] Yulianto, Agus, dan Devi.2016 “*Perancangan dan Realisasi Segway Menggunakan NI myRio*”, Tugas Akhir, Institute Teknologi Sepuluh Nopember, Surabaya.
- [12] Yuga Aditya P, 2016. *Implementasi Sensor Accelerometer Gyroscope Dan Magnetometer*, Teknik Elektro : Universitas Komputer Indonesia, <http://elib.unikom.ac.id/download.php?id=221180>

-----Halaman ini sengaja dik osongkan-----

## LAMPIRAN A

### A.1 Listing Program Stick Commander (Transmitter)

```
#include<Wire.h>
long previousMillis, currentMillis;
int xPin = A1;
int yPin = A0;
int a = 0;
int b = 0;
const int MPU_addr=0x68;
double AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
uint32_t timer;
double compAngleX, compAngleY;
int dataAwalX = 0;
int dataAwalY = 0;
int rangeX= 0;
int rangeY= 0;
#define degconvert 57.2957786

void setup() {
  pinMode(5, OUTPUT);
  pinMode(xPin, INPUT);
  pinMode(yPin, INPUT);
  // Set up MPU 6050:
  Wire.begin();
  #if ARDUINO >= 157
  Wire.setClock(400000UL);
  #else
  TWBR = ((F_CPU / 400000UL) - 16) / 2;
  #endif
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
  pinMode(2, INPUT_PULLUP);
  attachInterrupt(0, interruptnol, FALLING);
  delay(100);
```

```

//1) collect the data
Wire.beginTransmission(MPU_addr);
Wire.write(0x3B); // starting with register 0x3B
(ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr,14,true);
AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) &
0x3C (ACCEL_XOUT_L)
AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H)
& 0x3E (ACCEL_YOUT_L)
AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H)
& 0x40 (ACCEL_ZOUT_L)
Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) &
0x42 (TEMP_OUT_L)
GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H)
& 0x44 (GYRO_XOUT_L)
GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H)
& 0x46 (GYRO_YOUT_L)
GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) &
0x48 (GYRO_ZOUT_L)
//2) calculate pitch and roll
double roll = atan2(AcY, AcZ)*degconvert;
double pitch = atan2(-AcX, AcZ)*degconvert;
//3) set the starting angle to this pitch and roll
double gyroXangle = roll;
double gyroYangle = pitch;
double compAngleX = roll;
double compAngleY = pitch;
//start a timer
timer = micros();
}

void loop()
{
currentMillis = millis();
filter();
int xp = analogRead(xPin);
xp=xp+14;
xp = map(xp, 0, 1023, -10, 10);

```

```

int yp = analogRead(yPin);
yp=yp+5;
yp = map(yp, 0, 1023, -10, 10);
rangeX=compAngleX-dataAwalX;
rangeY=compAngleY-dataAwalY;
if(rangeX<0)
{
    rangeX=rangeX*-1;
    a=0;
    b=1;
}
else
{
    a=1;
    b=0;
}
if (digitalRead(2)==0)
{
    Serial.print("*");
    Serial.print(rangeY);
    Serial.print(",");
    Serial.print(a);
    Serial.print(",");
    Serial.print(b);
    Serial.print(",");
    Serial.print(rangeX);
    Serial.print(",");
    Serial.print(xp);
    Serial.print(",");
    Serial.print(yp);
    Serial.println("#");
    digitalWrite(5,HIGH);
    delay(100);
}
else
{
    Serial.print("*");
    Serial.print("0");
    Serial.print(",");

```

```

Serial.print("0");
Serial.print(",");
Serial.print("0");
Serial.print(",");
Serial.print("0");
Serial.print(",");
Serial.print(xp);
Serial.print(",");
Serial.print(yp);
Serial.println("#");
dataAwalX=0;
dataAwalY=0;
digitalWrite(5,LOW);
delay(100);
}
}

void filter()
{
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true);
  AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) &
  0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H)
  & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H)
  & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) &
  0x42 (TEMP_OUT_L)
  GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H)
  & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H)
  & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) &
  0x48 (GYRO_ZOUT_L)
  double dt = (double)(micros() - timer) / 1000000;
  timer = micros();
}
}

```

```

double roll = atan2(AcY, AcZ)*degconvert;
double pitch = atan2(-AcX, AcZ)*degconvert;
double gyroXrate = GyX/131.0;
double gyroYrate = GyY/131.0;
compAngleX = 0.99 * (compAngleX + gyroXrate * dt) + 0.01 *
roll;
compAngleY = 0.99 * (compAngleY + gyroYrate * dt) + 0.01 *
pitch;
}

void interruptnol()
{
dataAwalX=compAngleX;
dataAwalY=compAngleY;
}

```

### A.3 Dokumentasi

Bagian Dalam Stick Commander



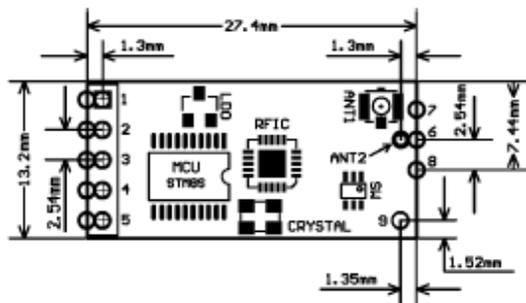


-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN B

### B.1 Datasheet HC-12

#### Product Dimensions



#### Definition of Pins

The HC-12 module can be patch soldered, or have a 2.54mm-spacing pin header attached and directly inserted onto the user's PCB. The module has nine pins in total, and one RF antenna socket (ANT1), with definitions as shown in the table below:

Pin	Definition	I/O direction	Notes
1	Vcc		Power supply input, DC3.2V-5.5V, with load capacity not less than 200mA. Note: if the module is working in the transmitting state for an extended time, it is suggested that a 1N4007 diode be connected in series if the supply voltage is greater than 4.5V, so as to avoid overheating the onboard LDO regulator
2	GND		Common ground
3	RxD	Input (weak pullup)	UART data input, TTL level. 1k resistor connected in series inside the module
4	TxD	Output	UART data output, TTL level. 1k resistor connected in series inside the module
5	SET	Input (10k pullup)	Parameter setting control pin, active low level. 1k resistor connected in series inside the module
6	ANT	Input/Output	433MHz antenna pin
7	GND		Common ground
8	GND		Common ground
9	NC		No connection, used in mechanical fixing, compatible with HC-11 module pin position
ANT1	ANT	Input/Output	IPEX20279-001E-03 antenna socket
ANT2	ANT	Input/Output	433MHz spring antenna solder eyelet

Pins 1-6 each have two bonding pads, with the outer half-hole bonding pads intended for patch soldering. When the inner bonding pad ANT2 of Pin 6 is used for connection, the spring antenna can be soldered here by hand. The inner round-hole bonding pads of Pins 1-5 may then be used to solder a 2.54mm-spacing pin header that can be plugged into a PCB socket.

with the corresponding relationship as shown in the table below:

Serial port baud rate	1200 bps	2400 bps	4800 bps	9600 bps	19,200 bps	38,400 bps	57,600 bps	115,200 bps
Baud rate in the air	5000bps		15,000bps		58,000bps		236,000bps	

To get the maximum communication distance, the serial port baud rate should be set to be low (1200bps or 2400bps). For short-time transmission of mass data, the serial port baud rate may be set high, but be aware that the communication distance will be reduced accordingly.

The receiving sensitivity of the module at different baud rates in the air is as shown in the table below:

Baud rate in the air	5000bps	15,000bps	58,000bps	236,000bps
Wireless receiving sensitivity	-117dBm	-112dBm	-107dBm	-100dBm

Generally, every time the receiving sensitivity is reduced by 6dB, the communication distance will be reduced by half.

When the "SET" pin of the module is pulled low, the serial port transparent transmission mode and other parameters can be set through AT commands (see the introduction in the following chapter for details).

**FU1** mode is a moderate power saving mode, with an idle working current of about 3.6mA. In this mode, the module can also be set to any of the eight serial port baud rates shown in the above table, but the baud rate in the air is a uniform 250,000bps.

**FU2** mode is an extreme power saving mode, with an idle working current of about 80uA. In this mode, the module only supports baud rates of 1200bps, 2400bps, and 4800bps, with the baud rate in the air uniform at 250,000bps. If the module is subsequently set to any other serial port baud rate, the module will not be able to conduct wireless communication normally.

When the module is set to FU2 mode, if the currently set baud rate exceeds 4800bps it will be automatically reduced to 4800bps. In FU2 mode, the sending time interval of data packets can not be too short, otherwise data will be lost. It is suggested that the sending time interval between data packets should be no less than 1 second.

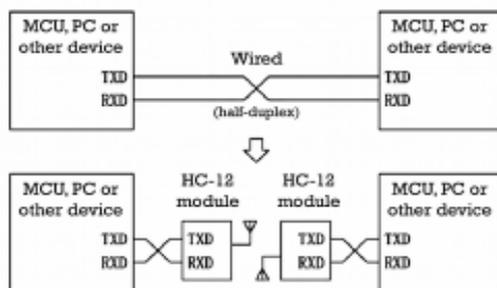
**FU4** mode is useful for maximum range, up to 1.8km. Only a single baud rate of 1200bps is supported, with the in the air baud rate reduced to 500bps for improved communication distance. This mode can only be used for small amounts of data (each packet should be 60 bytes or less), and the time interval between sending packets must not be too short (preferably no less than 2 seconds) in order to prevent loss of data.

The following table gives typical reference values for the various modes:

Mode	FU1	FU2	FU3	FU4	Remarks
Idle current	3.6mA	80uA	16mA	16mA	Average value
Transmission time delay	15-25ms	500ms	4-80ms	1000ms	Sending one byte
Loopback test time delay 1	31ms				Serial port baud rate 9600, sending one byte

## Wireless Serial Port Transparent Transmission

### (1) Simple introduction of working principle



As shown in the above diagram, two HC-12 modules can be used in place of physical wiring to replace a wired half-duplex serial communications link carrying TTL level signals. The left device sends serial port data to the module, and after the RxD port of the left module receives the serial port data, it will automatically send the data over the air via radio wave. The right module receives the data, and restores the serial port data originally sent by the left device and sends it out TxD. It is the same from right to left. Only a half-duplex link is available between modules, as they can not receive and send data over the air at the same time.

### (2) Serial port transparent transmission

The HC-12 module has four serial port transparent transmission modes, expressed as FU1, FU2, FU3, and FU4. In operation, these modes hide all the details of wireless communications from attached devices. The factory default working mode of the system is FU3 full-speed mode, and in this mode the baud rate in this air is automatically adjusted according to baud rate that the serial port has been set to. The usable communication distance will be the farthest at the lowest baud rate. Different modes can not transmit data to each other, and the user should select the optimal mode according to practical circumstances.

The modules are usually operated in pairs, with data transmitted by means of a half-duplex link. For successful wireless transmission, the transparent transmission mode, serial port baud rate, and wireless communication channel of the two paired modules must be set the same. The factory default module setting are: FU3, 9,600bps (8N1: 8 data bits, no parity, 1 stop bit), CH001 (433.4MHz), 20dBm power (100mW).

The number of bytes that can be continuously sent to the serial port of the module is unlimited in modes FU1 and FU3. However, considering ambient interference and other factors, if thousands of data bytes are sent continuously, some number of bytes may be lost. Therefore, the attached devices at each end of the link should have some sort of response and resending mechanism to avoid information loss.

### (3) The four serial port transparent transmission modes

When the HC-12 module leaves the factory, its default serial port transparent transmission mode is FU3. In this mode the module remains in full-speed state, with an idle current of about 16mA. The module automatically adjusts the baud rate of wireless transmission in the air according to the serial port baud rate,

Loopback test time delay 2	31ns				Serial port baud rate 9600, sending 10 bytes
Operating range at full power (20dBm)	100m	100m	600m at 9600bps 1000m at 1200bps 2400bps	1800m at 1200bps	Clear line of sight between modules under ideal conditions

Note: Loopback test time delay means the round trip time taken for data that is sent to the input (RXD pin) of one module, to begin to emerge from the output (TXD pin) of the same module, where a second (remote) module has been configured with the TXD and RXD pins connected together.

#### Module Parameter Setting AT Commands

AT commands are used to set module parameters and switch between module functions when the module is in command mode. After being set, these changes will become valid only after exiting from command mode. Parameters are stored in onboard non-volatile flash memory, so will not be lost when power is removed.

##### (1) Entering command mode

There are two ways to enter command mode:

1. while energized, pull Pin 5 ("SET") low, wait 40ms for command mode to engage
2. disconnect the power supply, connect Pin 5 ("SET") to GND, re-energize the module

Note: pin 5 has a 10k pullup resistor connected internally, allowing the pin to be driven by an open-collector output from an attached device.

Either of the above two methods will place the module in command mode ready to accept AT commands; releasing pin 5 ("SET") in either case exits from command mode. If the module settings have changed after exiting from command mode, it will be switched to the new settings within 80ms.

When the second method (pin 5 "SET" tied to ground before power is applied), the module always enters command mode with the serial port configured for 9600bps, 8 data bits, no parity, 1 stop bit, irrespective of any previously configured settings.

##### (2) Command instructions

- AT  
Test command. Send command "AT" to the module, and the module returns "OK".

- AT+BXXXX  
Change the serial port baud rate. The baud rate can be set to 1200bps, 2400bps, 4800bps, 9600bps, 19,200bps, 38,400bps, 57,600bps, or 115,200bps. The default value is 9600bps.

e.g: To set the serial port baud rate of the module to 19,200bps, send command "AT+B19200" to the module, and the module will return "OK+B19200". After exiting from command mode, the module will begin to communicate at 19,200bps.

- AT+CXXX  
Change wireless communication channel, selectable from 001 to 127 (for wireless channels exceeding 100, the communication distance cannot be guaranteed). The default value for the wireless channel is 001, with a working frequency of 433.4MHz. The channel stepping is 400KHz, and the working frequency of channel

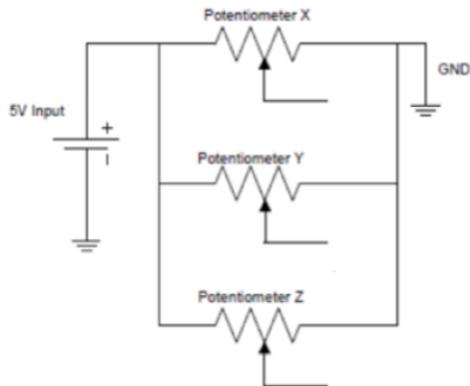
## B.2 Datasheet Joystick Specifications

- Directional movements are simply two potentiometers - one for each axis
- Compatible with Arduino interface
- The biaxial XY Joystick Module KY-023 applies ARDUINO
- Dimensions: 1.57 in x 1.02 in x 1.26 in (4.0 cm x 2.6 cm x 3.2 cm)
- 5 Pin
- Color: Black

### Pin Configuration

1. GND: ground
2. +5V: 5V DC
3. VRx: voltage proportional to x position
4. VRy: voltage proportional to y position
5. SW: switch pushbutton

### Schematic Diagram



## B.3 Datasheet MPU6050

### 5 Features

#### 5.1 Gyroscope Features

The triple-axis MEMS gyroscope in the MPU-60X0 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000^\circ/\text{sec}$
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Gyroscope operating current: 3.6mA
- Standby current: 5 $\mu$ A
- Factory calibrated sensitivity scale factor
- User self-test

#### 5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  and  $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500 $\mu$ A
- Low power accelerometer mode current: 10 $\mu$ A at 1.25Hz, 20 $\mu$ A at 5Hz, 60 $\mu$ A at 20Hz, 110 $\mu$ A at 40Hz
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

## 6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
<b>GYROSCOPE SENSITIVITY</b>						
Full-Scale Range	FS_SEL=0		±250		°/s	
	FS_SEL=1		±500		°/s	
	FS_SEL=2		±1000		°/s	
	FS_SEL=3		±2000		°/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)	
	FS_SEL=1		65.5		LSB/(°/s)	
	FS_SEL=2		32.8		LSB/(°/s)	
	FS_SEL=3		16.4		LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
<b>GYROSCOPE ZERO-RATE OUTPUT (ZRO)</b>						
Initial ZRO Tolerance	25°C		±20		°/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		°/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		°/s	
Linear Acceleration Sensitivity	Static		0.1		°/s/g	
<b>SELF-TEST RESPONSE</b>						
Relative	Change from factory trim	-14		14	%	1
<b>GYROSCOPE NOISE PERFORMANCE</b>						
Total RMS Noise	FS_SEL=0 DLPFCFG=2 (100Hz)		0.05		°/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to 10Hz		0.033		°/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		°/s/√Hz	
<b>GYROSCOPE MECHANICAL FREQUENCIES</b>						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
<b>LOW PASS FILTER RESPONSE</b>						
	Programmable Range	5		256	Hz	
<b>OUTPUT DATA RATE</b>						
	Programmable	4		8,000	Hz	
<b>GYROSCOPE START-UP TIME</b>						
ZRO Setting (from power-on)	DLPFCFG=0 to ±1% of Final		30		ms	

- Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

#### 6.4 Electrical Specifications, Continued

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
<b>SERIAL INTERFACE</b> SPI Operating Frequency, All Registers Read/Write	MPU-6000 only, Low Speed Characterization		100 ±10%		kHz	
	MPU-6000 only, High Speed Characterization		1 ±10%		MHz	
SPI Operating Frequency, Sensor and Interrupt Registers Read Only	MPU-6000 only		20 ±10%		MHz	
ƒC Operating Frequency	All registers, Fast-mode			400	kHz	
	All registers, Standard-mode			100	kHz	
<b>ƒC ADDRESS</b>	AD0 = 0		1101000			
	AD0 = 1		1101001			
<b>DIGITAL INPUTS (SDI/SDA, AD0, SCL/SCL, FSYNC, /CS, CLKIN)</b>						
V <sub>IH</sub> , High Level Input Voltage	MPU-6000	0.7*VDD			V	
	MPU-6050	0.7*VLOGIC			V	
V <sub>IL</sub> , Low Level Input Voltage	MPU-6000			0.3*VDD	V	
	MPU-6050			0.3*VLOGIC	V	
C <sub>i</sub> , Input Capacitance			< 5		pF	
<b>DIGITAL OUTPUT (SDO, INT)</b>						
V <sub>OH</sub> , High Level Output Voltage	R <sub>LOAD</sub> =1MΩ; MPU-6000	0.9*VDD			V	
	R <sub>LOAD</sub> =1MΩ; MPU-6050	0.9*VLOGIC			V	
V <sub>OL1</sub> , LOW-Level Output Voltage	R <sub>LOAD</sub> =1MΩ; MPU-6000			0.1*VDD	V	
	R <sub>LOAD</sub> =1MΩ; MPU-6050			0.1*VLOGIC	V	
V <sub>OLINT1</sub> , INT Low-Level Output Voltage	OPEN=1, 0.3mA sink current			0.1	V	
Output Leakage Current	OPEN=1		100		nA	
t <sub>INT</sub> , INT Pulse Width	LATCH_INT_EN=0		50		μs	

#### 6.5 Electrical Specifications, Continued

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

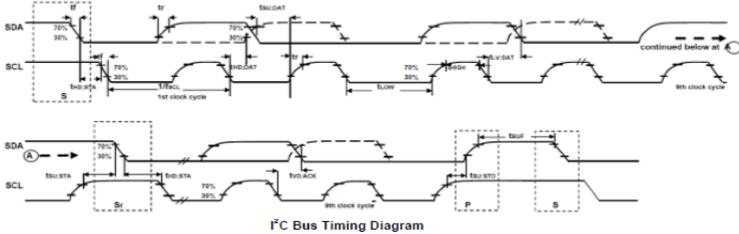
Parameters	Conditions	Typical	Units	Notes
<b>Primary ƒC I/O (SCL, SDA)</b>				
V <sub>IL</sub> , LOW-Level Input Voltage	MPU-6000	-0.5 to 0.3*VDD	V	
V <sub>IH</sub> , HIGH-Level Input Voltage	MPU-6000	0.7*VDD to VDD + 0.5V	V	
V <sub>HYH</sub> , Hysteresis	MPU-6000	0.1*VDD	V	
V <sub>IL</sub> , LOW Level Input Voltage	MPU-6050	-0.5V to 0.3*VLOGIC	V	
V <sub>IH</sub> , HIGH-Level Input Voltage	MPU-6050	0.7*VLOGIC to VLOGIC + 0.5V	V	
V <sub>HYH</sub> , Hysteresis	MPU-6050	0.1*VLOGIC	V	
V <sub>OL1</sub> , LOW-Level Output Voltage	3mA sink current	0 to 0.4	V	
I <sub>OL</sub> , LOW-Level Output Current	V <sub>OL</sub> = 0.4V	3	mA	
	V <sub>OL</sub> = 0.6V	5	nA	
Output Leakage Current		100	nA	
t <sub>FD</sub> , Output Fall Time from V <sub>IHTIME</sub> to V <sub>ILTMAX</sub>	C <sub>D</sub> bus capacitance in pF	20+0.1C <sub>D</sub> to 250	ns	
C <sub>D</sub> , Capacitance for Each I/O pin		< 10	pF	
<b>Auxiliary ƒC I/O (AUX_CL, AUX_DA)</b>	MPU-6050: AUX_VDDIO=0			
V <sub>IL</sub> , LOW-Level Input Voltage		-0.5V to 0.3*VLOGIC	V	
V <sub>IH</sub> , HIGH-Level Input Voltage		0.7*VLOGIC to VLOGIC + 0.5V	V	
V <sub>HYH</sub> , Hysteresis		0.1*VLOGIC	V	
V <sub>OL1</sub> , LOW-Level Output Voltage	VLOGIC > 2V; 1mA sink current	0 to 0.4	V	
V <sub>OL2</sub> , LOW-Level Output Voltage	VLOGIC < 2V; 1mA sink current	0 to 0.2*VLOGIC	V	
I <sub>OL</sub> , LOW-Level Output Current	V <sub>OL</sub> = 0.4V	1	mA	
	V <sub>OL</sub> = 0.6V	1	mA	
Output Leakage Current		100	nA	
t <sub>FD</sub> , Output Fall Time from V <sub>IHTIME</sub> to V <sub>ILTMAX</sub>	C <sub>D</sub> bus capacitance in pF	20+0.1C <sub>D</sub> to 250	ns	
C <sub>D</sub> , Capacitance for Each I/O pin		< 10	pF	

### 6.7 I<sup>2</sup>C Timing Characterization

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

Parameters	Conditions	Min	Typical	Max	Units	Notes
<b>I<sup>2</sup>C TIMING</b>						
f <sub>SCL</sub> , SCL Clock Frequency	I <sup>2</sup> C FAST-MODE			400	kHz	
t <sub>HD,STA</sub> , (Repeated) START Condition Hold Time		0.6			µs	
t <sub>LOW</sub> , SCL Low Period		1.3			µs	
t <sub>HIGH</sub> , SCL High Period		0.6			µs	
t <sub>SETUP,STA</sub> , Repeated START Condition Setup Time		0.6			µs	
t <sub>HD,DATA</sub> , SDA Data Hold Time		0			µs	
t <sub>SETUP,DATA</sub> , SDA Data Setup Time		100			ns	
t <sub>r</sub> , SDA and SCL Rise Time	C <sub>b</sub> bus cap. from 10 to 400pF	20+0.1C <sub>b</sub>		300	ns	
t <sub>f</sub> , SDA and SCL Fall Time	C <sub>b</sub> bus cap. from 10 to 400pF	20+0.1C <sub>b</sub>		300	ns	
t <sub>FREE</sub> , Bus Free Time Between STOP and START Condition		0.6			µs	
t <sub>SETUP,FREE</sub> , Bus Free Time Between STOP and START Condition		1.3			µs	
C <sub>b</sub> , Capacitive Load for each Bus Line			< 400		pF	
t <sub>VD,DATA</sub> , Data Valid Time				0.9	µs	
t <sub>VD,ACK</sub> , Data Valid Acknowledge Time				0.9	µs	

Note: Timing Characteristics apply to both Primary and Auxiliary I<sup>2</sup>C Bus

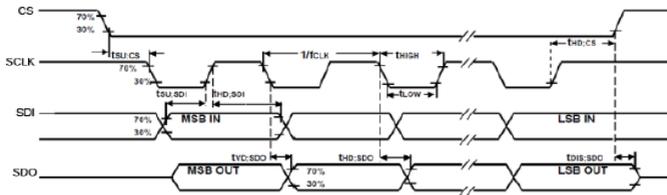


I<sup>2</sup>C Bus Timing Diagram

### 6.8 SPI Timing Characterization (MPU-6000 only)

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C, unless otherwise noted.

Parameters	Conditions	Min	Typical	Max	Units	Notes
<b>SPI TIMING</b>						
f <sub>CLK</sub> , SCLK Clock Frequency				1	MHz	
t <sub>LOW</sub> , SCLK Low Period		400			ns	
t <sub>HIGH</sub> , SCLK High Period		400			ns	
t <sub>SETUP,CS</sub> , CS Setup Time		8			ns	
t <sub>HD,CS</sub> , CS Hold Time		500			ns	
t <sub>SETUP,SDI</sub> , SDI Setup Time		11			ns	
t <sub>HD,SDI</sub> , SDI Hold Time		7			ns	
t <sub>VD,SDO</sub> , SDO Valid Time	C <sub>load</sub> = 20pF			100	ns	
t <sub>HD,SDO</sub> , SDO Hold Time	C <sub>load</sub> = 20pF	4			ns	
t <sub>DIS,SDO</sub> , SDO Output Disable Time				10	ns	



SPI Bus Timing Diagram

### 7.15 Internal Clock Generation

The MPU-60X0 has a flexible clocking scheme, allowing a variety of internal or external clock sources to be used for the internal synchronous circuitry. This synchronous circuitry includes the signal conditioning and ADCs, the DMP, and various control circuits and registers. An on-chip PLL provides flexibility in the allowable inputs for generating this clock.

Allowable internal sources for generating the internal clock are:

- An internal relaxation oscillator
- Any of the X, Y, or Z gyros (MEMS oscillators with a variation of  $\pm 1\%$  over temperature)

Allowable external clocking sources are:

- 32.768kHz square wave
- 19.2MHz square wave

Selection of the source for generating the internal synchronous clock depends on the availability of external sources and the requirements for power consumption and clock accuracy. These requirements will most likely vary by mode of operation. For example, in one mode, where the biggest concern is power consumption, the user may wish to operate the Digital Motion Processor of the MPU-60X0 to process accelerometer data, while keeping the gyros off. In this case, the internal relaxation oscillator is a good clock choice. However, in another mode, where the gyros are active, selecting the gyros as the clock source provides for a more accurate clock source.

Clock accuracy is important, since timing errors directly affect the distance and angle calculations performed by the Digital Motion Processor (and by extension, by any processor).

There are also start-up conditions to consider. When the MPU-60X0 first starts up, the device uses its internal clock until programmed to operate from another source. This allows the user, for example, to wait for the MEMS oscillators to stabilize before they are selected as the clock source.

### 7.16 Sensor Data Registers

The sensor data registers contain the latest gyro, accelerometer, auxiliary sensor, and temperature measurement data. They are read-only registers, and are accessed via the serial interface. Data from these registers may be read anytime. However, the interrupt function may be used to determine when new data is available.

For a table of interrupt sources please refer to Section 8.

### 7.17 FIFO

The MPU-60X0 contains a 1024-byte FIFO register that is accessible via the Serial Interface. The FIFO configuration register determines which data is written into the FIFO. Possible choices include gyro data, accelerometer data, temperature readings, auxiliary sensor readings, and FSYNC input. A FIFO counter keeps track of how many bytes of valid data are contained in the FIFO. The FIFO register supports burst reads. The interrupt function may be used to determine when new data is available.

For further information regarding the FIFO, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.

### 7.18 Interrupts

Interrupt functionality is configured via the Interrupt Configuration register. Items that are configurable include the INT pin configuration, the interrupt latching and clearing method, and triggers for the interrupt. Items that can trigger an interrupt are (1) Clock generator locked to new reference oscillator (used when switching clock sources); (2) new data is available to be read (from the FIFO and Data registers); (3) accelerometer event interrupts; and (4) the MPU-60X0 did not receive an acknowledge from an auxiliary sensor on the secondary I<sup>2</sup>C bus. The interrupt status can be read from the Interrupt Status register.

For further information regarding interrupts, please refer to the MPU-60X0 Register Map and Register Descriptions document.

For information regarding the MPU-60X0's accelerometer event interrupts, please refer to Section 8.

### 7.19 Digital-Output Temperature Sensor

An on-chip temperature sensor and ADC are used to measure the MPU-60X0 die temperature. The readings from the ADC can be read from the FIFO or the Sensor Data registers.

## 8 Programmable Interrupts

The MPU-60X0 has a programmable interrupt system which can generate an interrupt signal on the INT pin. Status flags indicate the source of an interrupt. Interrupt sources may be enabled and disabled individually.

**Table of Interrupt Sources**

Interrupt Name	Module
FIFO Overflow	FIFO
Data Ready	Sensor Registers
I <sup>2</sup> C Master errors: Lost Arbitration, NACKs	I <sup>2</sup> C Master
I <sup>2</sup> C Slave 4	I <sup>2</sup> C Master

For information regarding the interrupt enable/disable registers and flag registers, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document. Some interrupt sources are explained below.

### 9.2 I<sup>2</sup>C Interface

I<sup>2</sup>C is a two-wire interface comprised of the signals serial data (SDA) and serial clock (SCL). In general, the lines are open-drain and bi-directional. In a generalized I<sup>2</sup>C interface implementation, attached devices can be a master or a slave. The master device puts the slave address on the bus, and the slave device with the matching address acknowledges the master.

The MPU-60X0 always operates as a slave device when communicating to the system processor, which thus acts as the master. SDA and SCL lines typically need pull-up resistors to VDD. The maximum bus speed is 400 kHz.

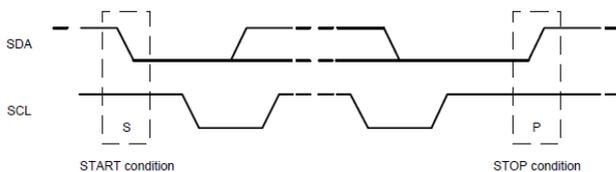
The slave address of the MPU-60X0 is b110100X which is 7 bits long. The LSB bit of the 7 bit address is determined by the logic level on pin AD0. This allows two MPU-60X0s to be connected to the same I<sup>2</sup>C bus. When used in this configuration, the address of the one of the devices should be b1101000 (pin AD0 is logic low) and the address of the other should be b1101001 (pin AD0 is logic high).

### 9.3 I<sup>2</sup>C Communications Protocol

#### *START (S) and STOP (P) Conditions*

Communication on the I<sup>2</sup>C bus starts when the master puts the START condition (S) on the bus, which is defined as a HIGH-to-LOW transition of the SDA line while SCL line is HIGH (see figure below). The bus is considered to be busy until the master puts a STOP condition (P) on the bus, which is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH (see figure below).

Additionally, the bus remains busy if a repeated START (Sr) is generated instead of a STOP condition.

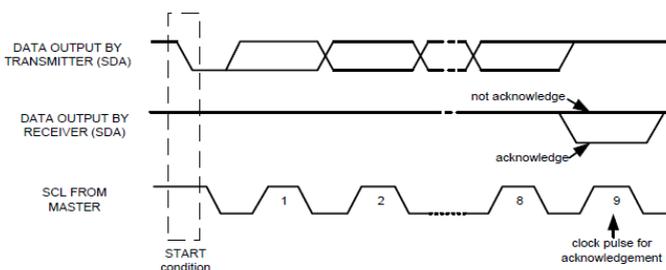


**START and STOP Conditions**

*Data Format / Acknowledge*

I<sup>2</sup>C data bytes are defined to be 8-bits long. There is no restriction to the number of bytes transmitted per data transfer. Each byte transferred must be followed by an acknowledge (ACK) signal. The clock for the acknowledge signal is generated by the master, while the receiver generates the actual acknowledge signal by pulling down SDA and holding it low during the HIGH portion of the acknowledge clock pulse.

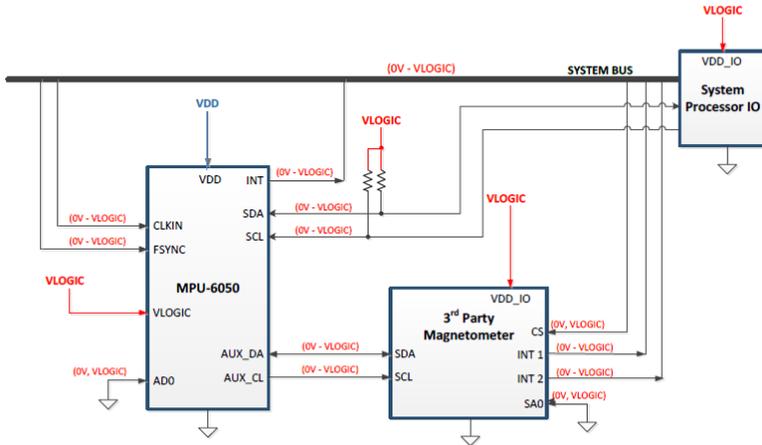
If a slave is busy and cannot transmit or receive another byte of data until some other task has been performed, it can hold SCL LOW, thus forcing the master into a wait state. Normal data transfer resumes when the slave is ready, and releases the clock line (refer to the following figure).



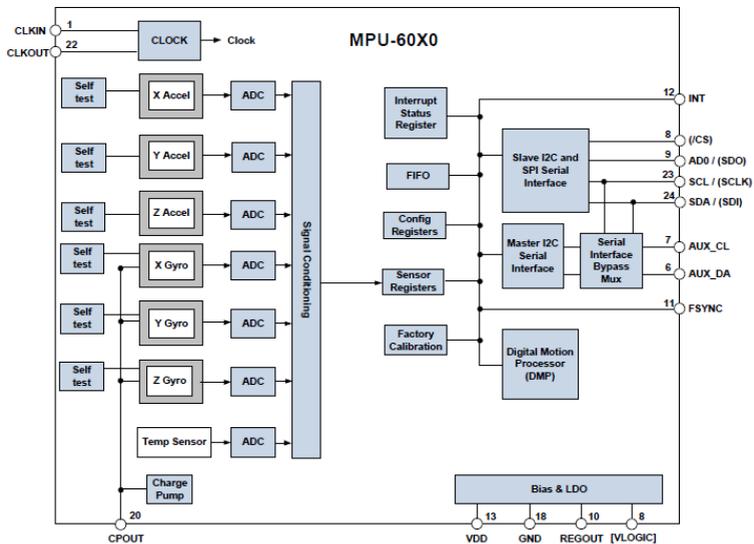
**Acknowledge on the I<sup>2</sup>C Bus**

### 10.3 Logic Levels Diagram for AUX\_VDDIO = 0

The figure below depicts a sample circuit with a third party magnetometer attached to the auxiliary I<sup>2</sup>C bus. It shows logic levels and voltage connections for AUX\_VDDIO = 0. Note: Actual configuration will depend on the auxiliary sensors used.



### 1.5 Block Diagram



Note: Pin names in round brackets ( ) apply only to MPU-6000  
Pin names in square brackets [ ] apply only to MPU-6050

## B.4 Datasheet Bosst Converter MT3608



**AEROSEMI**

**MT3608**

**High Efficiency 1.2MHz  
2A Step Up Converter**

### FEATURES

- Integrated 80mΩ Power MOSFET
- 2V to 24V Input Voltage
- 1.2MHz Fixed Switching Frequency
- Internal 4A Switch Current Limit
- Adjustable Output Voltage
- Internal Compensation
- Up to 28V Output Voltage
- Automatic Pulse Frequency Modulation Mode at Light Loads
- up to 97% Efficiency
- Available in a 6-Pin SOT23-6 Package

### APPLICATIONS

- Battery-Powered Equipment
- Set-Top Boxed
- LCD Bias Supply
- DSL and Cable Modems and Routers
- Networking cards powered from PCI or PCI express slots

### GENERAL DESCRIPTION

The MT3608 is a constant frequency, 6-pin SOT23 current mode step-up converter intended for small, low power applications. The MT3608 switches at 1.2MHz and allows the use of tiny, low cost capacitors and inductors 2mm or less in height. Internal soft-start results in small inrush current and extends battery life.

The MT3608 features automatic shifting to pulse frequency modulation mode at light loads. The MT3608 includes under-voltage lockout, current limiting, and thermal overload protection to prevent damage in the event of an output overload. The MT3608 is available in a small 6-pin SOT-23 package.

### TYPICAL APPLICATION

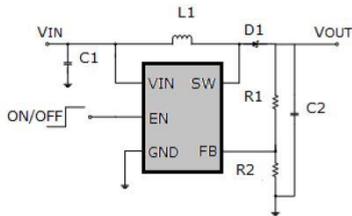


Figure 1. Basic Application Circuit

### Efficiency

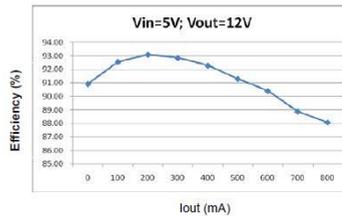
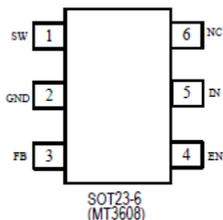


Figure 2. Efficiency Curve

## ABSOLUTE MAXIMUM RATINGS

IN, EN voltages .....	-0.3V to 26V	SW Voltage .....	-0.3V to 30V
Operating Temperature.....	-40°C to +85°C	Storage Temperature Range	-65°C to 150°C
FB Voltages .....	-0.3V to 6V	Peak SW Sink and Source Current .....	4A
Junction Temperature .....	160°C	Lead Temperature (Soldering, 10s) ...	+300°C

## PACKAGE/ORDER INFORMATION



## PIN DESCRIPTION

PIN	NAME	FUNCTION
1	SW	Power Switch Output. SW is the drain of the internal MOSFET switch. Connect the power inductor and output rectifier to SW. SW can swing between GND and 28V.
2	GND	Ground Pin
3	FB	Feedback Input. The FB voltage is 0.6V. Connect a resistor divider to FB.
4	EN	Regulator On/Off Control Input. A high input at EN turns on the converter, and a low input turns it off. When not used, connect EN to the input supply for automatic startup.
5	IN	Input Supply Pin. Must be locally bypassed.
6	NC	NC



## APPLICATION INFORMATION

### Setting the Output Voltage

The internal reference VREF is 0.6V (Typical). The output voltage is divided by a resistor divider, R1 and R2 to the FB pin. The output voltage is given by

$$V_{OUT} = V_{REF} \times \left(1 + \frac{R_1}{R_2}\right)$$

### Inductor Selection

The recommended values of inductor are 4.7 to 22μH. Small size and better efficiency are the major concerns for portable device, such as MT3608 used for mobile phone. The inductor should have low core loss at 1.2MHz and low DCR for better efficiency. To avoid inductor saturation current rating should be considered.

### Capacitor Selection

Input and output ceramic capacitors of 22μF are recommended for MT3608 applications. For better voltage filtering, ceramic capacitors with low ESR are recommended. X5R and X7R types are suitable because of their wider voltage and temperature ranges.

### Diode Selection

Schottky diode is a good choice for MT3608 because of its low forward voltage drop and fast reverse recovery. Using Schottky diode can get better efficiency. The high speed rectification is also a good characteristic of Schottky diode for high switching frequency. Current rating of the diode must meet the root mean square of the peak current and output average current multiplication as following :

$$I_D(RMS) \approx \sqrt{I_{OUT} \times I_{PEAK}}$$

The diode's reverse breakdown voltage should be larger than the output voltage.

### Layout Consideration

For best performance of the MT3608, the following guidelines must be strictly followed.

- Input and Output capacitors should be placed close to the IC and connected to ground plane to reduce noise coupling.
- The GND should be connected to a strong ground plane for heat sinking and noise protection.
- Keep the main current traces as possible as short and wide.
- SW node of DC-DC converter is with high frequency voltage swing. It should be kept at a small area.
- Place the feedback components as close as possible to the IC and keep away from the noisy devices.

## B.5 Datasheet Arduino Nano



# Technical Specification



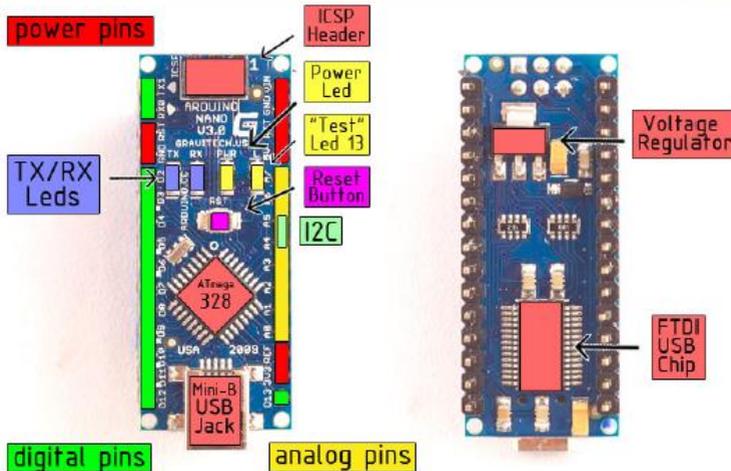
Arduino Nano 3.0 (ATmega328): [schematic](#), [Eagle files](#).

Arduino Nano 2.3 (ATmega168): [manual](#) (pdf), [Eagle files](#). Note: since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

### Summary

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

### the board



## Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

## Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the [EEPROM library](#)); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

## Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I<sup>2</sup>C: 4 (SDA) and 5 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega168 ports](#).

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

## RIWAYAT HIDUP PENULIS



Nama : Nova Agustina  
TTL : Mojokerto, 20 Agustus  
1996  
: Perempuan  
Agama : Islam  
Alamat Rumah : Tarik 05/03, Sidoarjo  
Telp/HP : +6285706673175  
E-mail : novaagustina@gmail.com  
Hobi : Hiking

### RIWAYAT PENDIDIKAN

- 2001 – 2002 : TK Aisyah Kletek Taman
- 2002 – 2008 : SDN Kletek
- 2008 – 2011 : SMP Negeri 1 Taman
- 2011 – 2014 : SMA Negeri 1 Krian
- 2014 – 2017 : Bidang Studi Komputer Kontrol, Program D3 Teknik Elektro, ITS

### PENGALAMAN KERJA

- Kerja Praktek di PT PJB Unit Pembangkitan Gresik, Jawa Timur

### PENGALAMAN ORGANISASI

- Staff RISTEK Himpunan Mahasiswa D3 Teknik Elektro ITS 2015-2016.
- Kepala Divisi RISTEK Himpunan Mahasiswa D3 Teknik Elektro ITS 2016-2017.

### SERTIFIKASI DAN PENGHARGAAN

- Connecting Osiloskop (2015)