TUGAS AKHIR – TE 145561

TELEMETERING KEBOCORAN PIPA PADA DISTRIBUSI AIR DENGAN KOMUNIKASI *ETHERNET*

Farrah Fadilah
NRP 10311400010019

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng
Onie Meiyanto, S.Pd.

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

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

FINAL PROJECT – **TE 145561**

TELEMETERING PIPELINE LEAKS ON WATER DISTRIBUTION WITH ETHERNET COMMUNICATION

Farrah Fadilah
NRP 10311400010019

Supervisor
Ir. Josaphat Pramudijanto, M.Eng
Onie Meiyanto, S.Pd.

Automation Electrical Engineering Department
Faculty of Vocation
Institut Teknologi Sepuluh Nopember
Surabaya 2018

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

PERNYATAAN KEASLIAN
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Telemetring Kebocoran Pipa Pada Distribusi Air dengan Komunikasi *Ethernet***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

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

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

Surabaya, **10 Januari 2018**

Farrah Fadilah
NRP 10311400010019

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

TELEMETRING KEBOCORAN PIPA PADA DISTRIBUSI AIR DENGAN KOMUNIKASI *ETHERNET*

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui,

Dosen Pembimbing I,	Dosen Pembimbing II,
Ir. Josaphat Pramudijanto, M.Eng.	Onie Meiyanto, S.Pd.
NIP. 19621005 199003 1 003	NIP. 19850501 201101 1 008

SURABAYA
JANUARI, 2018

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

TELEMETRING KEBOCORAN PIPA PADA DISTRIBUSI AIR DENGAN KOMUNIKASI ETHERNET

Nama Mahasiswa : Farrah Fadilah
NRP : 10311400010019
Pembimbing I : Ir. Josaphat Pramudijanto, M.Eng.
NIP : 19621005 199003 1 003
Pembimbing II : Onie Meiyanto, S.Pd.
NIP : 19850501 201101 1 008

ABSTRAK

Kebocoran pipa dapat menyebabkan kerugian finansial bagi pelanggan sistem distribusi air. Pada umumnya untuk mendeteksi lokasi kebocoran pipa pada jaringan distribusi air menggunakan metode manual, yakni dengan melihat genangan air di atas instalasi pipa yang terpasang ataupun dengan adanya pelaporan kebocoran pipa dari pelanggan.

Maka dari itu pada Tugas Akhir ini dibuatlah *prototype* telemetering kebocoran pipa. Kebocoran dapat diketahui dengan menggunanakan metode perbandingan pengukuran debit air pada dua sensor *water flowmeter*. Metode ini menggunakan sensor *water flowmeter* yang ditempatkan sebelum dan setelah kebocoran. Kebocoran pipa disimulasikan dengan 2 kran air, ada 2 titik kebocoran, agar air yang terbuang dapat dialirkan pada sumber air (tandon) dan air dapat bersikulasi. Data yang didapat dari pembacaan sensor dikirim ke komputer menggunakan jaringan komunikasi *Ethernet* dan ditampilkan pada PC dengan *software LabVIEW*.

Setelah dilakukan pengujian, hasilnya menunjukkan bahwa kedua sensor memiliki nilai selisih antara aliran air masuk dan aliran air keluar, menunjukkan adanya kebocoran. Hasilnya untuk mendeteksi kebocoran pipa berhasil dilakukan. Telemetering dengan metode perbandingan debit air pada sensor *water flowmeter* dapat mengetahui seberapa besar kebocoran debit air pada pipa, yaitu nilai selisih pembacaan sensor sebesar 24 L/hour dengan persentase kebocoran sebesar 6,9%.

Kata Kunci : Telemetering, Sistem Distribusi Air, *Flowmeter*, Kebocoran Pipa.

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

TELEMETRING PIPELINE LEAKS ON WATER DISTRIBUTION WITH ETHERNET COMMUNICATION

Student's Name	**: Farrah Fadilah**
Regristration Number	**: 10311400010019**
***Supervisor* I**	**: Ir. Josaphat Pramudijanto, M.Eng.**
ID *Number*	**: 19621005 199003 1 003**
***Supervisor* II**	**: Onie Meiyanto, S.Pd.**
ID *Number*	**: 19850501 201101 1 008**

ABSTRACT

Pipe leaks can cause financial losses for customers of water distribution systems. In general, to detect the location of pipeline leaks in the water distribution network using manual method, ie by looking at standing water on the installation of the pipe installed or with the reporting of leaking pipes from customers.

Therefore in this Final Project prototype made telemetering leakage pipe. Leakage can be known by using the method of comparison of measurement of water debit on two water flowmeter sensors. This method uses water flowmeter sensors placed before and after leakage. The leakage of the pipe is simulated with 2 water faucets, there are 2 points of leakage, so that the wasted water can be drained on the water source (tandon) and the water can circulate. Data obtained from sensor readings are sent to the computer using an Ethernet communication network and displayed on a PC with LabVIEW software.

After testing, the results show that both sensors have a difference value between inlet flow and outlet flow, indicating leakage. The result to detect pipeline leak successfully. Telemetering with the method of water discharge comparison on the water flowmeter sensor can find out how much leakage of water debit in the pipe, the value of sensor readout difference of 24 L / hour with leakage percentage of 6.9%.

Keywords : Telemetering, Water Distribution System, Flowmeter, Pipe Leakage.

-----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 Program Studi Elektronika Industri, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

TELEMETRING KEBOCORAN PIPA PADA DISTRIBUSI AIR DENGAN KOMUNIKASI *ETHERNET*

Tugas Akhir ini merupakan syarat untuk menyelesaikan mata kuliah dan memperoleh nilai pada Tugas Akhir. Dengan selesainya Tugas Akhir ini penulis menyampaikan terimakasih yang sebesar-besarnya kepada :

1. Kedua orangtua yang selalu memberi dukungan, doa, dan memberi perhatian kepada penulis.
2. Bapak Joko Susila, ST., MT selaku Ketua Departemen Teknik Elektro Otomasi, FV-ITS.
3. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku dosen pembimbing
4. Bapak Onie Meiyanto, S.Pd. selaku dosen pembimbing dari BLK
5. 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, 10 Januari 2018

Penulis

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

DAFTAR ISI

HALAMAN

HALAMAN JUDUL i
PERNYATAAN KEASLIAN TUGAS AKHIR v
HALAMAN PENGESAHAN vii
ABSTRAK ix
ABSTRACT xii
KATA PENGANTAR xiii
DAFTAR ISI xv
DAFTAR GAMBAR xvii
DAFTAR TABEL xix
BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Permasalahan 2
1.3. Batasan Masalah 2
1.4. Tujuan 2
1.5. Sistematika Laporan 2
1.6. Relevansi 3
BAB II TEORI DASAR 5
2.1. Sistem Distribusi Air Bersih Secara Umum 5
2.2. Arduino Uno R3 5
2.3. *Water Flowmeter Sensor* 7
2.4. *Ethernet Shield* 7
2.5. *LabVIEW* 8
BAB III PERANCANGAN DAN PEMBUATAN ALAT 11
3.1. Diagram Fungsional Sistem 11
3.2. Perancangan Perangkat Keras (*Hardware*) 12
3.2.1. Perancangan *Prototype* Simulasi Kebocoran Pipa 12
3.2.2. Rangkaian *Water Flowmeter Sensor* 13
3.2.3. Koneksi *Ethernet Shield* dengan Arduino Uno 13
3.3. Perancangan Perangkat Lunak (*Software*) 14
3.3.1. Pemrograman Pembacaan Sensor *Flowmeter* 14
3.3.2. Perancangan *Software LabVIEW* 15
BAB IV PENGUJIAN DAN ANALISA 19
4.1. Pengujian *Input*/*Output* Arduino Uno 19
4.2. Kalibrasi *Water Flowmeter* Sensor 20
4.3. Pengujian Komunikasi *Ethernet* 21
4.3.1. Pengujian pada *Board* Arduino Uno 22
4.3.2. Pengujian pada *Software LabVIEW* 24
4.4. Pengujian Secara Keseluruhan 26
BAB V PENUTUP 29

5.1. Kesimpulan 29
5.2. Saran 29
DAFTAR PUSTAKA 31
LAMPIRAN A PROGRAM A-1
A.1. Program *Ethernet* 1 ..... A-1
A.2. Program *Ethernet* 2 ..... A-2
A.3. Program *Water Flowmeter Sensor* A-3
LAMPIRAN B *DATASHEET* B-1
B.1. *Water Flowmeter Sensor* G3/4" B-1
B.2. Arduino *Ethernet Shield* V1 ..... B-3
B.2.1. Wiznet W5100 ..... B-7
B.3. Arduino Uno R3 ..... B-18
LAMPIRAN C TAMPILAN *LABVIEW* C-1
C.1. *Block Diagram* C-1
C.2. *Front Panel* C-2
C.3. Tampilan Grafik C-3
RIWAYAT HIDUP PENULIS D-1

DAFTAR GAMBAR

HALAMAN

Gambar 2.1. *Board* Arduino Uno R3 6
Gambar 2.2. *Water Flowmeter* Sensor 7
Gambar 2.3. *Ethernet Shield* 8
Gambar 2.4. Tampilan Pada *LabVIEW* 9
Gambar 3.1. Diagram Fungsional Alat Keseluruhan 11
Gambar 3.2. Perancangan *Prototype* Simulasi Kebocoran Pipa 12
Gambar 3.3. Koneksi *Water Flowmeter Sensor* Ke Arduino Uno ... 13
Gambar 3.4. Koneksi *Ethernet Shield* Pada Arduino Uno 14
Gambar 3.5. *Flowchart* Pembacaan Sensor *Flowmeter* 15
Gambar 3.6. Block Diagram Pada *LabVIEW* 16
Gambar 3.7. Desain *Front Panel* Pada *LabVIEW* 17
Gambar 4.1. Program Pada Arduino Uno ke-1 22
Gambar 4.2. *Command Prompt* Pada Laptop Dengan IP *address* Arduino Uno ke-1 23
Gambar 4.3. Program Pada Arduino Uno ke-2 23
Gambar 4.4. *Command Prompt* Pada Laptop Dengan IP *address* Arduino Uno ke-2 24
Gambar 4.5. Tampilan *LabVIEW* Pada Pengujian Sensor *Water Flowmeter* ke-1 25
Gambar 4.6. Tampilan *LabVIEW* Pada Pengujian Sensor *Water Flowmeter* ke-2 25
Gambar 4.7. Tampilan *LabVIEW* Pada Saat Keadaan Normal 26
Gambar 4.8. Tampilan *LabVIEW* Pada Saat Kran Air Ke-1 Dibuka 27
Gambar 4.9. Tampilan *LabVIEW* Pada Saat Kran Air Ke-2 Dibuka 27
Gambar 4.10. Tampilan *LabVIEW* Saat Kedua Kran Air Dibuka ... 27

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

DAFTAR TABEL

HALAMAN

Tabel 4.1. Hasil Pengukuran Per *Pin* Saat *Active High*..... 19
Tabel 4.2. Hasil Pengukuran Per *Pin* Saat *Active Low*..... 20
Tabel 4.3. Hasil Kalibrasi Sensor *Water Flowmeter*..... 21
Tabel 4.4. Data Pengujian Kedua Buah Sensor Pada *LabVIEW*..... 25
Tabel 4.5. Data Pengujian Keseluruhan 28

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

BAB I
PENDAHULUAN

1.1. Latar Belakang

Dewasa ini, kebutuhan air bersih untuk masyarakat semakin meningkat seiring berjalannya waktu. Tidak hanya wilayah perkotaan saja yang membutuhkan air bersih, namun pada wilayah pedesaan pun mengalami peningkatan. Tentunya hal ini mengharuskan jaringan pendistribusian air dialirkan sampai ke desa-desa terpencil sekalipun. Pada jaringan pendistribusian ini pasti membutuhkan penyambungan pipa yang banyak sekali. Pada penyambungan pipa inilah sering terjadi kebocoran pipa. Yang dimana kebocoran pipa tersebut sulit untuk terdeteksi. Sehingga biasanya informasi kebocoran pipa air didapatkan dari keluhan masyarakat, selain itu dapat didapatkan dari petugas yang melakukan pengecekan pipa secara rutin. Pengecekan kebocoran pipa dilakukan pada malam hari, hal ini dikarenakan pada malam hari tekanan air dalam pipa sangat tinggi sehingga kebocoran pipa dapat mudah diketahui. Meskipun begitu, mengetahui letak kebocoran pipa masih dinilai kurang cepat. Sehingga untuk melakukan perbaikan dapat dinilai sangat lamban. Terlambatnya perbaikan kebocoran mengakibatkan air bersih yang mengalir terbuang percuma. Tentu hal ini dapat disayangkan, karena jika air pada penampungan suatu daerah tidak banyak akan mengurangi pasokan air untuk masyarakat.

Agar pasokan air cukup untuk masyarakat dan air tidak terbuang percuma, maka perlu adanya sistem yang dapat mudah mengetahui kebocoran air yang terjadi pada pipa. Oleh karena itu, pada Tugas Akhir ini mengambil judul "Telemetring Kebocoran Pipa Pada Distribusi Air Dengan Komunikasi *Ethernet"*. Yakni pembuatan sebuah pengukuran debit air menggunakan 2 (dua) sensor *flowmeter* dengan melakukan perbandingan agar dapat mengetahui kebocoran pipa. Perbandingan debit air tersebut nantinya sebagai data persentase kebocoran air yang ada pada pipa. Data tersebut dikirim oleh Arduino menggunakan komunikasi Ethernet yang akan ditampilkan pada *software LabVIEW*. Hal ini selain mempermudah petugas jaga, juga lebih efisien untuk mengetahui kebocoran air pada pipa, serta dapat memantau kondisi pipa dengan jarak jauh melalui *personal computer* pada *software LabVIEW* dengan menggunakan komunikasi Ethernet.

1.2. Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan utama adalah sulitnya mengetahui informasi kebocoran pipa.

1.3. Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari tugas akhir ini adalah :

1. Sensor yang digunakan sebagai pembaca debit air adalah sensor *water flowmeter* ¾" sebanyak 2 buah sensor.
2. Pengukuran yang dilakukan hanya terbatas pada pengukuran debit air saja.
3. Instalasi pipa yang digunakan sepanjang 537 cm.
4. Pengaplikasian hanya terbatas sistem distribusi air.

Dengan adanya batasan masalah ini diharapkan hasil akhir atau tujuan dari Tugas Akhir ini dapat dicapai dengan baik.

1.4. Tujuan

Pembuatan Telemetering Kebocoran Pipa Pada Distribusi Air Dengan Komunikasi *Ethernet* bertujuan untuk:

1. Merancang dan membuat prototype kebocoran pipa pada sistem distribusi air.
2. Merancang dan membuat perangkat lunak untuk mendeteksi aliran distribusi air.

1.5. 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, sistematika laporan, dan relevansi.

Bab II Teori Penunjang

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari pendistribusian air secara umum, Arduino Uno, sensor *flowmeter*, *Ethernet Shield*, dan *LabVIEW*.

Bab III Perancangan dan Pembuatan Alat

Bab ini membahas perancangan sistem *hardware* maupun *software* pada Telemetering Kebocoran Pipa Pada Distribusi Air Dengan

Komunikasi *Ethernet* berdasarkan teori penunjang pada Bab II.

Bab IV **Pengujian dan Analisa**

Bab ini membahas tentang pengukuran, pengujian, serta analisa terhadap prinsip kerja dan proses dari suatu alat yang dibuat.

Bab V **Penutup**

Bab ini menjelaskan tentang kesimpulan dari Tugas Akhir dan saran – saran untuk pengembangan alat ini lebih lanjut.

1.6. Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dengan pengembangan lebih lanjut dapat dijadikan sebagai referensi atau ide dalam kemajuan sistem telemetering pada jaringan distribusi air, khususnya pada telemetering kebocoran pipa. Dengan efisiensi monitoring kebocoran pipa melalui jarak jauh ini semoga dapat membantu mendeteksi kebocoran pipa pada jaringan distribusi air.

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

BAB II
TEORI PENUNJANG

2.1. Sistem Distribusi Air Bersih Secara Umum [1]

Sistem Penyediaan Air Bersih (SPAB) terdiri dari beberapa sistem dan sub sistem serta komponen atau elemen sistem yaitu, sumber air baku, system pengolahan air (IPA), sistem transmisi, dan sistem distribusi. Dalam masing-masing komponen sistem dapat terdiri dari beberapa sub-sistem atau elemen sistem.

Sistem Pengolahan Air (IPA) terdiri dari satuan operasi dan satuan proses yang mencakup unit penyaringan (screening), unit pre-sedimentasi, unit koagulasi dan flokulasi, unit sedimentasi, unit filtrasi, unit disinfeksi, unit reservoir, dan system penunjang dalam bentuk unit pembubuh bahan kimia, sistem pompa pembubuh dan sebagainya.

Rangkaian sistem pengolahan air atau konfigurasi sistem pengolahan air terdiri dari satu atau beberapa rangkaian proses pengolahan. Secara umum, rangkaian pengolahan air didefenisikan sebagai urutan-urutan beberapa proses pengolahan yang diperlukan untuk memisahkan material tertentu sebagai sasaran pengolahan air yang dimaksudkan. Apabila proses pengolahan air tersebut ditujukan untuk memisahkan beberapa sasaran material, maka dengan demikian, rangkaian proses pengolahan air mungkin akan terdiri dari beberapa proses yang masing-masing dirancang untuk dapat memisahkan material tertentu. Untuk mengembangkan suatu konfigurasi sistem pengolahan air yang sistematis, maka dalam hal ini diklasifikasikan sasaran-sasaran pengolahan sesuai ukuran material-material yang akan dipisahkan. Sebagai contoh, jika material yang akan dipisahkan berupa material tersuspensi (suspended materials, misalnya parameter kekeruhan), maka sasaran pengolahan akan diklasifikasikan sebagai “pemisah material tersuspensi” (suspended particulate removal). Bilamana material yang akan dipisahkan berupa material terlarut (dissolved materials, misalnya parameter Fe dan Mn), maka sasaran pengolahan akan diklasifikasikan sebagai “pemisah material terlarut” (dissolved-particulate removal).

2.2. Arduino Uno R3 [2]

Arduino merupakan papan rangkaian elektronik atau kit elektronik yang bersifat *open-source* yang didalamnya terdapat

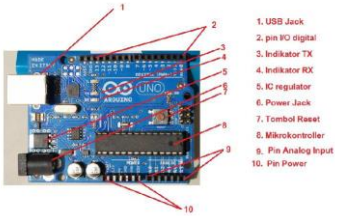
komponen utama yaitu sebuah mikrokontroler dengan jenis *Atmel AVR* dan menggunakan software sendiri untuk pemrograman.

Mikrokontroler sendiri adalah sebuah komputer kecil disuatu sikuit terpadu yang berisi tentang inti prosesor, memori dan input/output yang telah diprogram. Program disimpan dalam bentuk *Ferroelectric* RAM, *Nor Flash*, OTP ROM yang disertakan dalam chip. Mikrokontroler digunakan untuk aplikasi *embedded*, tidak seperti mikroprosesor yang digunakan dalam komputer pribadi. Fungsi dari mikrokontroler adalah untuk mengontrol produk atau perangkat secara otomatis seperti sistem kontrol mesin mobil, mesin kantor, alat-alat listrik, dan sistem *embedded* lainnya.

Salah satu jenis Arduino yang menggunakan mikrokontroler tipe *Atmel AVR* (8-bit) adalah Arduino Uno . Arduino Uno adalah papan mikrokontroler berbasiskan ATmega 328. Spesifikasinya adalah seperti berikut:

- Memiliki kecepatan *clock* mencapai 16MHz.
- Memiliki *Flash Memory* 32KB, 0.5KB diantaranya sebagai *bootloader.*
- Rentang *input* sumber tegangan 7-20 Volt, dan beroperasi pada tegangan 5 Volt.
- Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB.
- Memiliki SRAM (*Static Random Access Memory*) sebesar 2KB.
- Memiliki pin I/O digital sebanyak 14 pin dan 6 pin diantaranya digunakan sebagai *output* PWM (*Pulse Width Modulation*).
- Memiliki pin I/O analog sebanyak 6 pin.
- Arus DC per pin I/O 40 mA dan pada pada pin 3.3V 50mA.

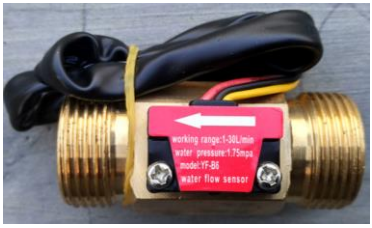
Letak pin pada *board* Arduino Uno dapat dilihat pada Gambar 2.1.



Gambar 2.1. *Board* Arduino Uno R3

2.3. *Water Flowmeter Sensor* [3]

Water flowmeter sensor adalah sensor aliran yang terdiri dari katub plastik, rotor air, dan sensor *hall effect*. Prinsip kerja sensor aliran air ini adalah dengan memanfaatkan fenomena efek hall. Efek hall merupakan suatu peristiwa berbeloknya aliran listrik (elektron) dalam pelat konduktor karena adanya pengaruh medan magnet terhadap partikel bermuatan yang bergerak. Ketika ada arus listrik yang mengalir pada efek hall yang ditempatkan dalam medan magnet yang arahnya tegak lurus arus listrik, pergerakan pembawa muatan akan berbelok ke salah satu sisi dan menghasilkan medan listrik. Lalu medan listrik akan terus membesar sehingga gaya Lorentz yang bekerja pada partikel menjadi sama dengan nol. Beda potensial antara sisi devais tersebut adalah potensial hall. Potensial hall ini berbanding lurus dengan medan magnet dan arus listrik yang melewati devais. Jadi semakin cepat magnet berputar akibat debit air yang membesar melewati devais, maka semakin besar arus listrik yang dihasilkan yang dapat dibaca oleh Arduino Uno sebagai *input* sinyal digital. Dari pembacaan sinyal digital oleh Arduino Uno ini diolah menjadi suatu besaran pada suatu objek. Pada kali ini sinyal digital akan dirubah menjadi besaran debit air dengan satuan L/hour. *Water flowmeter* ini hanya membutuhkan satu sinyal *output*, *vcc* 5Vdc dan *ground*. Bentuk *water flowmeter sensor* dapat dilihat pada Gambar 2.2 dibawah ini.



Gambar 2.2. *Water Flowmeter Sensor*

2.4. *Ethernet Shield* [4]

Ethernet Shield merupakan suatu perangkat yang dapat menambah kemampuan Arduino untukterhubung ke jaringan computer. *Ethernet shield* berbasiskan cip *Ethernet Wiznet* W5100. Cip *Ethernet Wiznet* W5100 ini menyediakan jaringan internet (IP) baik TCP dan UDP. Dan didukung oleh 4 soket koneksi yang simultan. Penggunaan perangkat ini mengacu pada *library Ethernet Shield* untuk penulisan programnya.

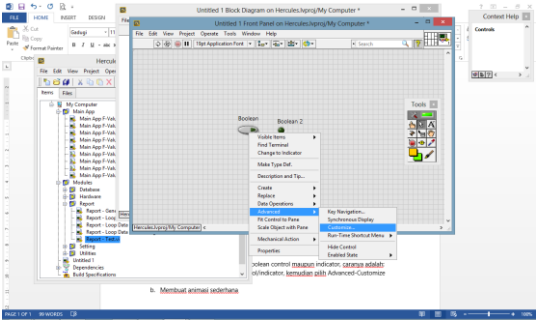
Pada *Ethernet Shield* terdapat sebuah slot *micro*-SD (*Secure Digital*), yang dapat digunakan untuk menyimpan file yang diakses melalui jaringan. *Board* Arduino dapat berkomunikasi dengan cip Winzet W5100 dan SD card menggunakan bus SPI (SERIAL Peripheral interface). Bus SPI menggunakan pin digital 11, 12, dan 13 pada Arduino Uno. Pin digital 10 digunakan untuk memilih W5100 dan pin digital 4 digunakan untuk memilih SD card. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk input/output umum ketika kita menggunakan ethernet shield. Karena W5100 dan SD card berbagi bus SPI, hanya salah satu yang dapat aktif pada satu waktu.

Gambar 2.3. *Ethernet Shield*

2.5. *LabVIEW* [5]

LabVIEW adalah sebuah software pemograman yang diproduksi oleh *National instruments* dengan konsep yang berbeda. Seperti bahasa pemograman lainnya yaitu C++, matlab atau *Visual basic*, *LabVIEW* juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa *LabVIEW* menggunakan Bahasa pemrograman berbasis *grafis* atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis *text*. Program *LabVIEW* dikenal dengan sebutan Vi atau *Virtual instruments* karena penampilan dan operasinya dapat meniru sebuah *instrument*. Pada *LabVIEW*, user pertama-tama membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan control adalah *knobs*, *push buttons*, dials dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah graphs, LEDs dan peralatan display lainnya.

Setelah menyusun user interface, lalu user menyusun blok diagram yang berisi kode-kode VIs untuk mengontrol *front panel*. *Software LabVIEW* terdiri dari tiga komponen utama, yaitu : *front panel*, Blok diagram dari Vi, *Control* dan *Functions Pallete*.



Gambar 2.4. Tampilan Pada *LabVIEW*

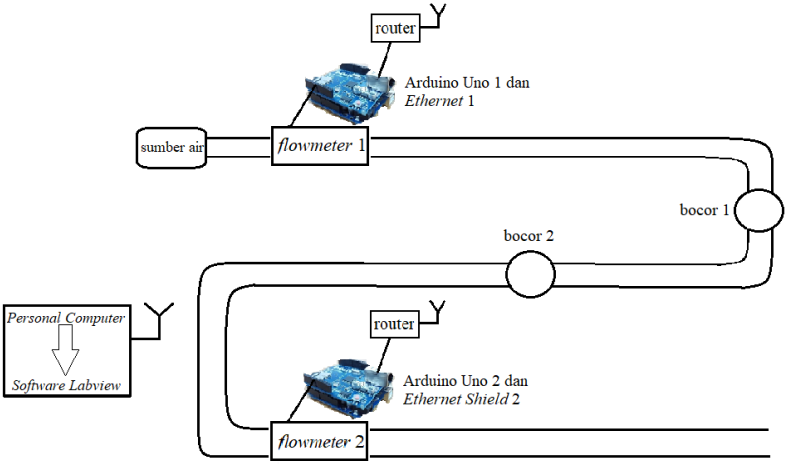
-----Halaman ini sengaja dikosongkan-----

BAB III
PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai perancangan dan pembuatan alat "Telemetring Kebocoran Pipa Pada Distribusi Air Dengan Komunikasi Ethernet*"*. Penjelasan diawali dengan penjelasan diagram fungsional sistem secara keseluruhan, kemudian perancangan perangkat keras dan diakhiri perencanaan perangkat lunak. Perancangan perangkat keras (*hardware*) meliputi perancangan *prototype* simulasi kebocoran pipa beserta perangkat elektriknya. Dan perancangan perangkat lunak (*software*) meliputi pemrograman arduino IDE pada arduino Uno untuk pengukuran debit air pada sensor *flowmeter* serta monitoring debit air dengan *interface LabVIEW*.

3.1. Diagram Fungsional Sistem

Perancangan sistem secara keseluruhan dalam pembuatan alat yang digunakan dapat diperlihatkan pada Gambar 3.1.



Gambar 3.1. Diagram Fungsional Alat Keseluruhan

Dari Gambar 3.1 dapat dipahami bahwa jika pompa air dan *power supply* diberi daya maka Arduino Uno akan menyala dan sensor *flowmeter* akan bekerja membaca debit air dari aliran air yang mengalir. Pembacaan sensor *flowmeter* diolah sedemikian rupa hingga didapatkan data debit air yang ditampilkan pada *personal*

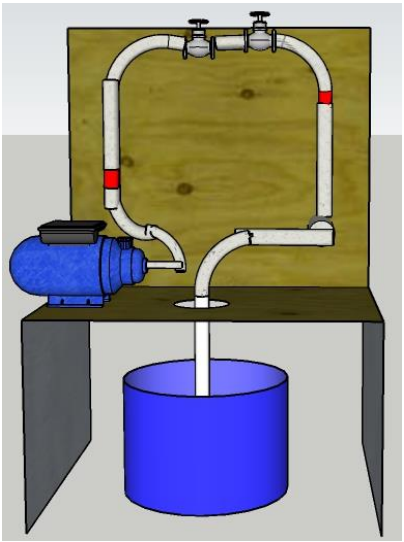
computer (PC) melalui perangkat lunak *LabVIEW* dengan komunikasi *ethernet compatible* Arduino.

3.2. Perancangan Perangkat Keras (*Hardware*)

Pada perancangan hardware pada tugas akhir ini yang dibahas terdiri dari perancangan *prototype* simulasi kebocoran pipa dan perancangan elektrik berupa rangkaian sensor beserta komunikasinya.

3.2.1. Perancangan *Prototype* Simulasi Kebocoran Pipa

Perancangan *prototype* simulasi kebocoran pipa ini dimaksudkan untuk jarak yang jauh dan dengan kondisi lingkungan yang tidak biasa.



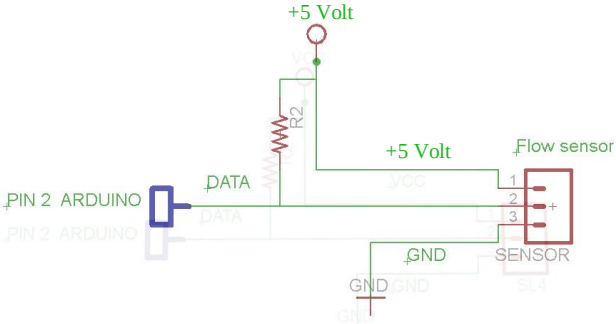
Gambar 3.2. Perancangan *Prototype* Simulasi Kebocoran Pipa

Rancangan *prototype* diatas menggunakan bak penampungan air dengan diameter 45 cm. Ukuran pipa yang digunakan sebesar ¾", sedangkan total panjang pipa yang digunakan sebesar 537 cm. Sensor *flowmeter* diletakkan pada 25% dan 75% dari total panjang pipa. Lebih tepatnya sensor *flowmeter* yang pertama diletakkan pada titik 134,3 cm, sedangkan sensor *flowmeter* yang kedua diletakkan pada titik 402,8 cm. Lalu sebagai simulator, maka akan diibaratkan terdapat 2 titik kebocaran. Dimana letak kebocoran tersebut

ditempatkan diantara 2 sensor *flowmeter*. Pada perancangan tersebut, kebocoran pipa digantikan dengan kran air. Jadi ketika salah satu kran air dibuka, maka akan diketahui kebocoran pipa yang dideteksi dari berkurangnya debit air. Titik kebocoran ditentukan sebanyak 2 titik dikarenakan untuk menguji seberapa akurat *prototype* yang akan dibuat.

3.2.2. Rangkaian *Water Flowmeter* Sensor

Pada Gambar 3.3. adalah skematik koneksi *Water Flowmeter* Sensor ke Arduino.



Gambar 3.3. Koneksi *Water Flowmeter Sensor* Ke Arduino Uno

Rangkaian Skematik *Water Flowmeter* Sensor Pin sinyal dari sensor dihubungkan dengan pin 2 Arduino yang dikonfigurasi sebagai input yang mana pin 2 tersebut akan memberikan cacahan naik ketika terjadi perubahan nilai. Pin sinyal dari sensor tersebut juga diberikan resistor pull-up dari Pin 5 Volt Arduino sendiri. Tujuan dari pemasangan resistor pull-up ini agar data yang diterima dari sensor memiliki nilai yang valid.

3.2.3. Koneksi *Ethernet Shield* dengan Arduino Uno

Pada perangkat komunikasi digunakan *Ethernet Shield* yang telah *compatible* dengan Arduino Uno. Sehingga pemasangan *Ethernet Shield* pada Arduino Uno hanya dengan digabungkan pada bagian atas Arduino Uno saja. Gabungan antara Arduino Uno dan *ethernet shield* sering dinamakan Arduino Web Server. Gabungan *Ethernet shield* yang dipasang di atas Arduino Uno dapat dilihat pada Gambar 3.4.

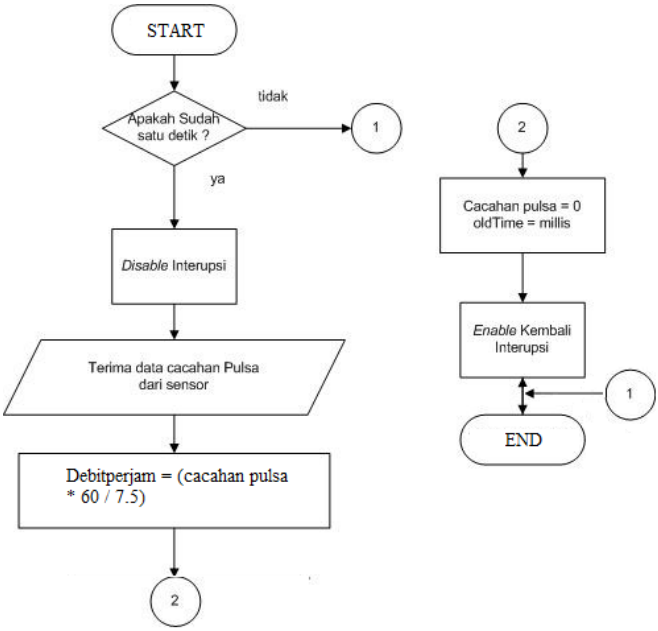
Gambar 3.4. Koneksi *Ethernet Shield* Pada Arduino Uno

3.3. Perancangan Perangkat Lunak (*Software*)

Pada perancangan perangkat *software* pada tugas akhir ini yang dibahas terdiri dari pemrograman pengukuran debit air sensor *flowmeter* pada software Arduino IDE, dan perancangan *software LabVIEW*.

3.3.1. Pemrograman Pembacaan Sensor *Flowmeter*

Pada Gambar 3.5 dapat dilihat alur kerja dari pembacaan sensor *flowmeter*. Pada pembacaan ini terdapat dua buah interupsi yang digunakan yaitu interupsi eksternal pada pin 2 arduino untuk menerima data dari sensor dan interupsi timer millis() untuk memungkinkan sistem dapat mengkonversikan data dari sensor setiap satu detik berlalu.



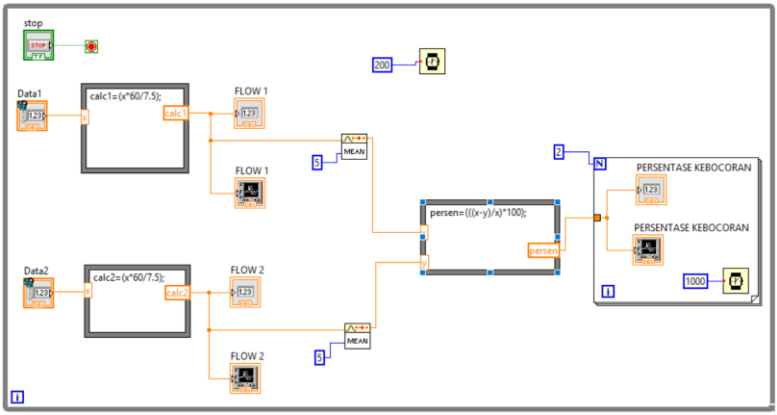
Gambar 3.5. *Flowchart* Pembacaan Sensor *Flowmeter*.

3.3.2. Perancangan *Software LabVIEW*

Pada perancangan *software LabVIEW* yang pertama dirancang yaitu membuat *block diagram* setelah itu merancang *front panel*. *Block diagram* berisi *source code* yang berfungsi sebagai instruksi untuk *front panel*. Sedangkan pada *front panel* sendiri mengandung *control* dan indikator untuk membangun sebuah VI (*Virtual Instruments*), menjalankan program dan men*debug* program.

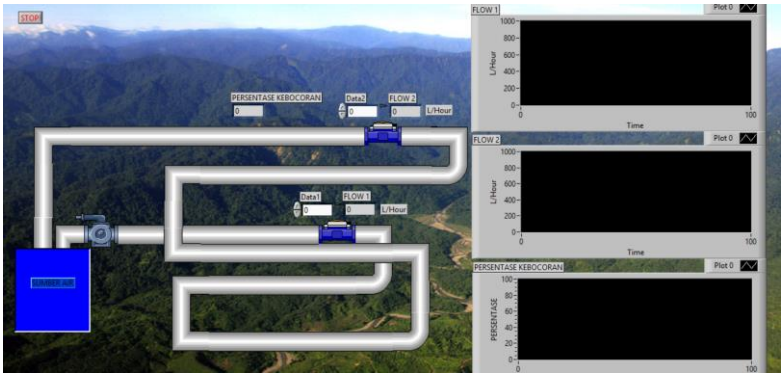
Pada perancangan *block diagram* (Gambar 3.6) bagian kiri terdapat *function palatte numeric* dengan tanda nama Data1 dan Data2 yang berguna untuk menampilkan pembacaan 2 sensor per 200 ms, ditunjukkan pada *function palatte* dengan *icon* jam, yang dikirim melalui *Ethernet*. Pada bagian kanan Data1 dan Data2 terdapat kotak berwarna abu-abu yang disebut dengan *formula node*, fungsi dari *function palatte* ini adalah untuk menuliskan rumus konversi sinyal digital menjadi besaran debit air (L/hour). Lalu pada bagian sebelah kanan *formula node* disambungkan pada *function palatte numeric* dan grafik sebagai hasil perhitungan debit air dengan tanda nama FLOW 1 dan FLOW 2. Nilai yang didapatkan dari hasil perhitungan lalu diambil 5 sampel data untuk diambil rata-rata.

Setelah mendapatkan nilai rata-rata 5 sampel data maka akan dilakukan perhitungan persentase kebocoran pada *function palatte formula node.* Setelah didapatkan nilai perhitungan persentase kebocoran, disambungkan pada *function palatte for loop* yang didalamnya terdapat *function palatte numeric*, grafik, dan *wait* untuk menampilkan nilai perhitungan dan grafik per 1000 ms.



Gambar 3.6. *Block Diagram* Pada *LabVIEW*

Sedangkan perancangan pada *front panel* (Gambar 3.7) dibuat gambar instalasi pipa yang dilengkapi dengan *water flowmeter sensor* dimana pada masing-masing sensor terdapat keterangan diatasnya. Pada bagian bawah tanda nama Data1 dan Data2 merupakan hasil dari pembacaan sensor, telah dijelaskan pada penjelasan sebelumnya. Lalu pada bagian bawah tanda nama FLOW 1 dan FLOW 2 merupakan hasil dari perhitungan konversi sinyal digital menjadi besaran debit air dengan satuan L/hour. Lalu hasil perhitungan persentase kebocoran ditampilakan pada bagian kiri Data2. Sedangkan gambar grafik diletakkan pada bagian kanan secara parallel, yang dimana telah terdapat tanda nama pada bagian atas gambar grafik.



Gambar 3.7. Desain *Front Panel* Pada *LabVIEW*

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

BAB IV
PENGUJIAN DAN ANALISA

Dalam membuat suatu sistem, pengujian dan analisa sangat diperlukan. Pengujian dan analisa bertujuan untuk mengetahui system bekerja sesuai rencana atau belum. Dari hasil pengujian dan analisadapat diketahui kelemahan-kelemahan dari sistem, sehingga dapat dilakukan perbaikan, pengembangan, dan penyempurnaan sistem.

Dalam bab ini dibahas pengujian serta analisa dari perancangan sistem yang telah dibuat pada tugas akhir. Pengujian dilakukan secara parsial terlebih dahulu, tujuannya adalah untuk mengetahui kinerja *hardware* setiap unit. Setelah mengetahui respon kinerja *hardware* setiap unit, kemudian akan dilakukan pengujian integrasi sistem secara keseluruhan. Adapun beberapa pengujian yang telah dilakukan adalah sebagai berikut.

4.1. Pengujian *Input/Output* Arduino Uno

Pengujian ini dilakukan pada rangkaian board Arduino Uno. Pengujian dilakukan untuk mengetahui tegangan *output* pada setiap *pin* arduino jika diberi *input high* dan *input low* dengan menggunakan voltmeter. Hasil pengukuran pada tiap pin dari *board* Arduino Uno dapat dilihat pada Tabel 4.1 dan 4.2.

Tabel 4.1. Hasil Pengukuran Per *Pin* Saat *Active High*

Pin Analog	Tegangan (Volt)	*Pin* Digital	Tegangan (Volt)
Pin A0	5,01	*Pin* 0	5,02
Pin A1	5,01	*Pin* 1	5,01
Pin A2	5,01	*Pin* 2	5,01
Pin A3	5,01	*Pin* 3	5,01
Pin A4	5,01	*Pin* 4	5,01
Pin A5	5,01	*Pin* 5	5,01
		Pin 6	5,01
		Pin 7	5,01
		Pin 8	5,01
		Pin 9	5,01
		Pin 10	5,01
		Pin 11	5,01
		Pin 12	5,01
		Pin 13	5,01

Dari hasil Tabel 4.1 dapat disimpulkan bahwa ketika board Arduino Uno diberi *input high* pada tiap *pin* maka tegangan yang dihasilkan bernilai rata-rata 5,01 Volt yang artinya tegangan yang dikeluarkan telah maksimal dan pin tersebut dapat berfungsi sebagai supply untuk beban yang diinginkan karena output board Arduino Uno bernilai *high* dengan range tegangan 2,4-5,5 Volt.

Tabel 4.2. Hasil Pengukuran Per *Pin* Saat *Active Low*

Pin Analog	Tegangan (mV)	*Pin* Digital	Tegangan (mV)
Pin A0	0,8	*Pin* 0	10,0
Pin A1	0,8	*Pin* 1	5,9
Pin A2	0,9	*Pin* 2	1,2
Pin A3	0,6	*Pin* 3	1,3
Pin A4	0,7	*Pin* 4	1,2
Pin A5	0,5	*Pin* 5	1,1
		Pin 6	1,1
		Pin 7	1,2
		Pin 8	1,3
		Pin 9	1,1
		Pin 10	1,1
		Pin 11	1,1
		Pin 12	1,2
		Pin 13	1,2

Dari hasil Tabel 4.2 dapat disimpulkan bahwa ketika *board* Arduino Uno diberi *input low* pada tiap *pin* maka tegangan yang dihasilkan bernilai rata-rata 1,76 mV yang artinya tegangan yang dikeluarkan sangat kecil dan *pin* tersebut dapat berfungsi sebagai *ground* karena *output board* Arduino Uno bernilai *low* dengan *range* tegangan 0–800 mV.

4.2. Kalibrasi *Water Flowmeter* Sensor

Sensor *water flowmeter* digunakan untuk mendeteksi laju aliran air yang akan menentukan jumlah debit air yang mengalir pada instalasi pipa yang telah dibuat. Sebelum dilakukan pengukuran kebocoran pada instalasi pipa, maka dibutuhkan kalibrasi untuk menentukan keakuratan sensor tersebut. Sensor ini dikalibrasi menggunakan gelas ukur standar dengan ketelitian 25 ml dan langkah kalibrasinya adalah sebagai berikut :

- Hubungkan rangkaian pengujian sensor *water flowmeter*

- Masukkan air pada masukan *flowmeter* (dilihat dari arah anak panah aliran air) dalam periode waktu tertentu
- Tampung air yang keluar pada *flowmeter* menggunakan gelas ukur standar
- Bersamaan dengan berhenti masukkan air, catat angka pada serial monitor arduino
- Lalu ukur volume air pada gelar ukur
- Penunjukkan angka pada gelas ukur dan pembacaan sensor merupakan debit air yang melewati *flowmeter*
- Hitung persentase kesalahan pembacaan sensor
- Ulangi langkah awal untuk kalibrasi sensor ke-2

Tabel 4.3. Hasil Kalibrasi Sensor *Water Flowmeter*

T (min)	Gelas ukur (L)	*Flowmeter* 1 (L)	*Flowmeter* 2 (L)	Kesalahan (%)
1	1,5	1,3	1,3	0,86667
2	2,5	2,2	2,2	0,88
3	3,5	3,4	3,4	0,97143
4	4,5	4,4	4,4	0,97778
5	5,5	5,3	5,3	0,96364
6	6,5	6,4	6,4	0,98642
7	7,5	7,2	7,3	0,96667
8	8,5	8,2	8,2	0,96471
9	9,5	9,1	9,1	0,96789
10	10,5	10,4	10,3	0,98571
11	11,5	11,3	11,3	0,98261
12	12,5	12,4	12,4	0,992

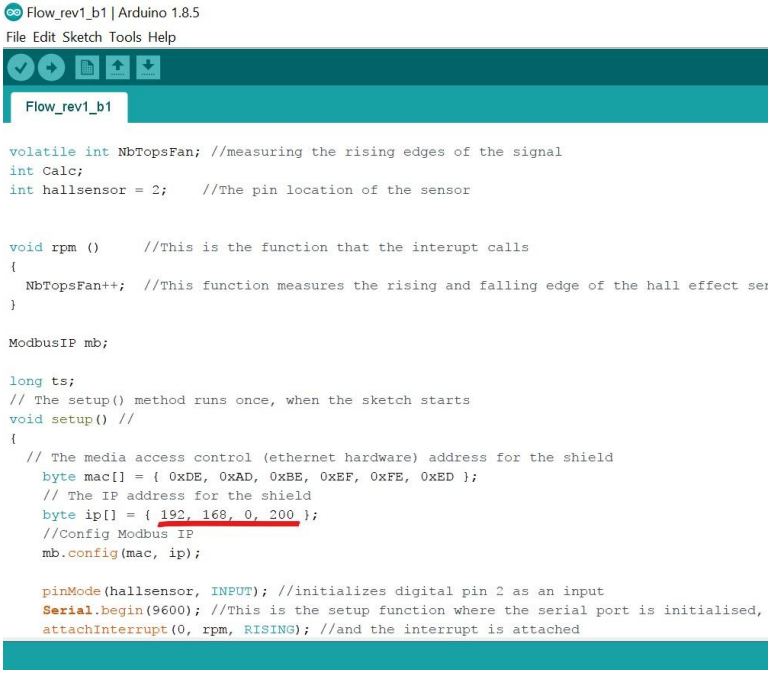
Berdasarkan hasil pengukuran kalibrasi oleh sensor *water flowmeter* dengan gelas ukur hampir menunjukkan angka yang sama. Data yang diperoleh dari keseluruhan sensor menunjukkan hasil yang linier dan stabil. Dikatakan stabil karena setiap menit sensor menunjukkan respon yang hampir sama.

4.3. Pengujian Komunikasi *Ethernet*

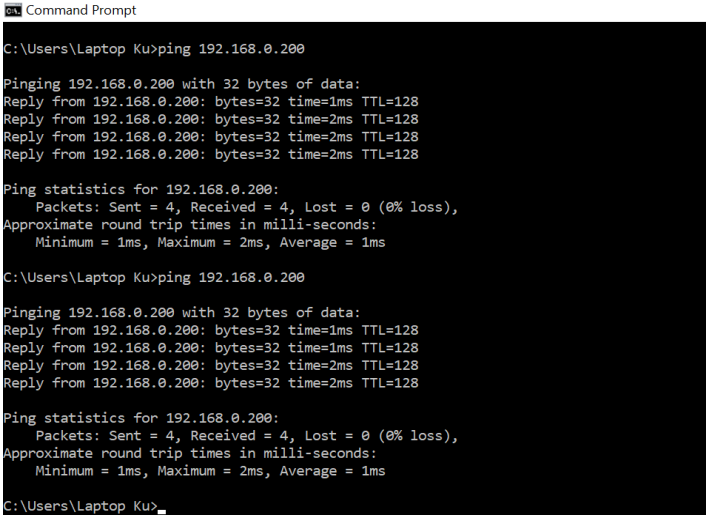
Pengujian ini dilakukan pada modul *Ethernet shield* yang telah dipasang pada bagian atas *board* Arduino Uno. Dalam hal ini pengujian komunikasi dibagi menjadi dua, yaitu pengujian komunikasi *Ethernet* pada *board* Arduino Uno dan pengujian komunikasi *Ethernet* dengan *software LabVIEW*.

4.3.1. Pengujian pada *Board* Arduino Uno

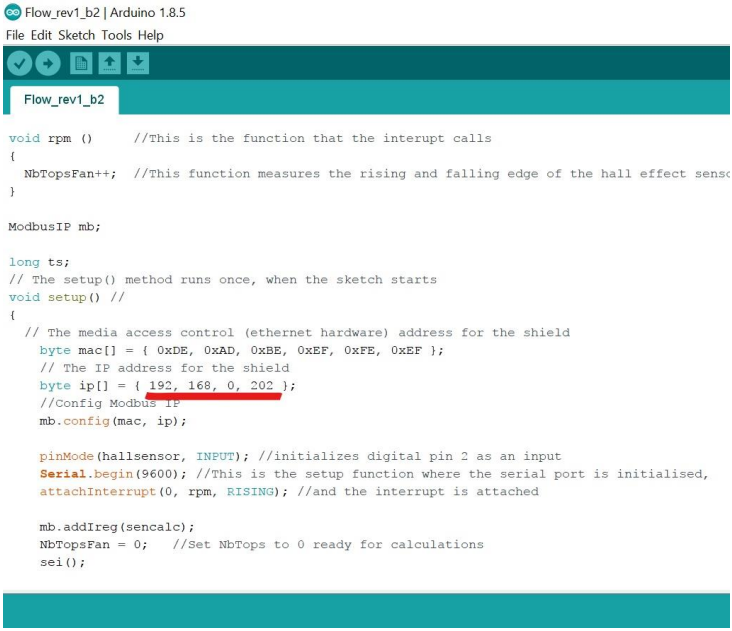
Sebelum menggunakan komunikasi *Ethernet*, sebaiknya dilakukan pengujian komunikasi *Ethernet* pada *board* Arduino Uno terlebih dahulu. Pengujian ini dilakukan dengan cara menyamakan *setting* IP (*Internet Protocol*) *address* yang telah diprogram pada *board* Arduino Uno, dapat dilihat pada Gambar 4.1 dan Gambar 4.3 yang bergaris bawah merah, dengan *setting* IP *address* pada *Ethernet shieild*. *Setting* IP *address* pada *Ethernet shield* dapat diketahui melalui *command prompt* pada suatu *notebook*/laptop (Gambar 4.2 dan Gambar 4.4).



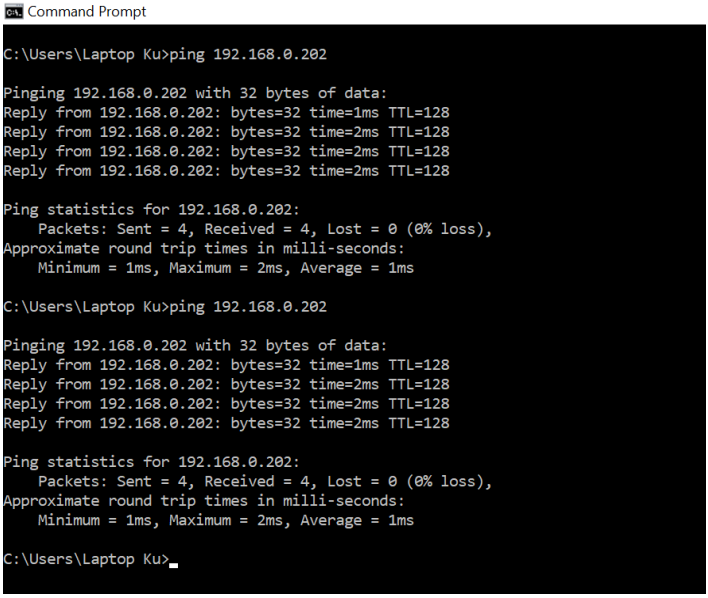
Gambar 4.1. Program Pada Arduino Uno ke-1



Gambar 4.2. *Command Prompt* Pada Laptop Dengan IP *address* Arduino Uno ke-1



Gambar 4.3. Program Pada Arduino Uno ke-2



Gambar 4.4. *Command Prompt* Pada Laptop Dengan IP *address* Arduino Uno ke-2

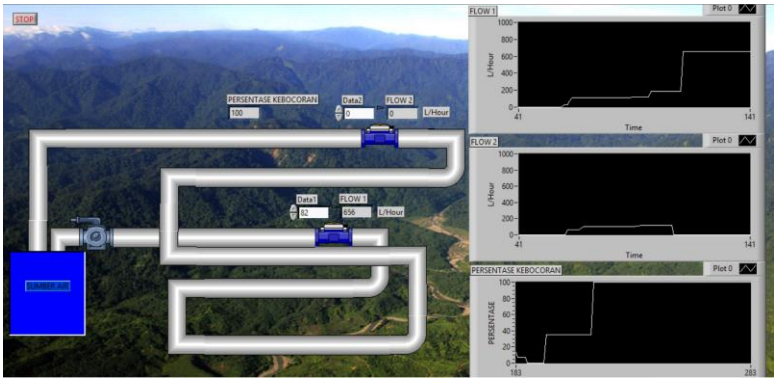
Dapat dilihat pada 4 gambar diatas menunjukkan bahwa *Ethernet shield* yang tersambung pada laptop telah siap digunakan. Ketika mengirim perintah *ping* nomor IP *address* pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP *address* telah bekerja, dapat dilihat pada Gambar 4.2 dan 4.4.

4.3.2. Pengujian pada *Software LabVIEW*

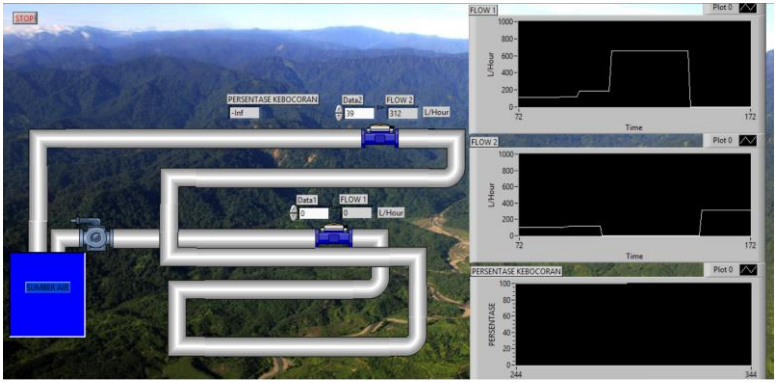
Setelah pengujian komunikasi *Ethernet* pada Arduino Uno, pengujian komunikasi *Ethernet* perlu dilakukan pada *software LabVIEW* untuk mengetahui apakah *Ethernet shield* dapat mengirimkan data pada *software LabVIEW* atau tidak.

Pengujian ini dilakukan dengan cara parsial pada rangkaian sensor *water flowmeter* ke-1 dan rangkaian sensor *water flowmeter* ke-2. Setelah rangkaian sensor tersambung, alirkan air pada sensor. Sembari mengalirkan air pada sensor, amati tampilan pada *software LabVIEW*. Setelah itu catat nilai yang terbaca pada tampilan

LabVIEW. Lakukan proses yang sama pada rangkaian sensor *water flowmeter* ke-2.



Gambar 4.5. Tampilan *LabVIEW* Pada Pengujian Sensor *Water Flowmeter* ke-1



Gambar 4.6. Tampilan *LabVIEW* Pada Pengujian Sensor *Water Flowmeter* ke-2

Tabel 4.4. Data Pengujian Kedua Buah Sensor pada *LabVIEW*

FLOW 1 (L/hour)	Data1 (bit)	FLOW 2 (L/hour)	Data2 (bit)
656	82	0	0
0	0	312	39

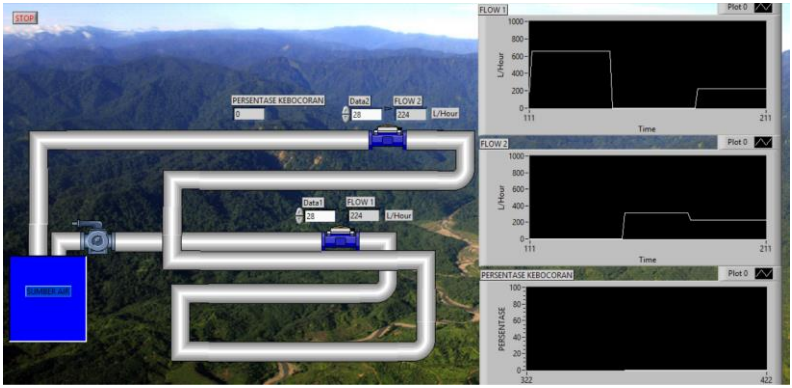
Dapat dilihat pada Gambar 4.5 didapatkan hasil data pada Tabel 4.4, kolom Data1 bernilai 82 dan FLOW 1 bernilai 656. Lalu

pada kolom Data2 bernilai 39 dan FLOW 2 bernilai 312. Pengujian komunikasi *Ethernet* pada *LabVIEW* telah tersambung. Hal ini dibuktikan pada Gambar 4.5 dan Gambar 4.6 bahwa saat sensor *water flowmeter* bekerja, *software LabVIEW* dapat menampilkan pembacaan sensor.

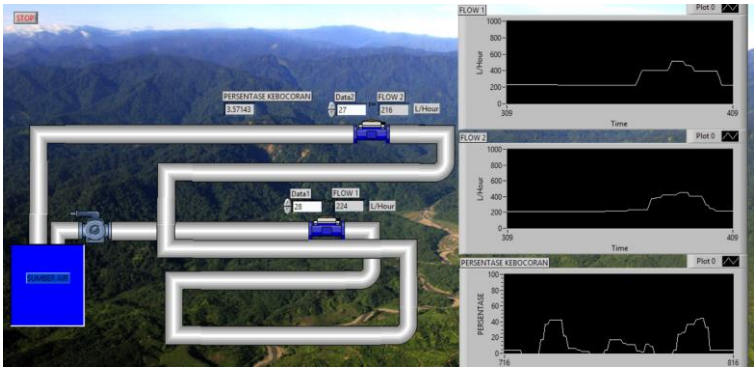
4.4. Pengujian Secara Keseluruhan

Pengujian secara keseluruhan merupakan gabungan dari beberapa pengujian yang telah dilakukan dan pengujian kebocoran pipa pada *prototype* simulasi distribusi air.

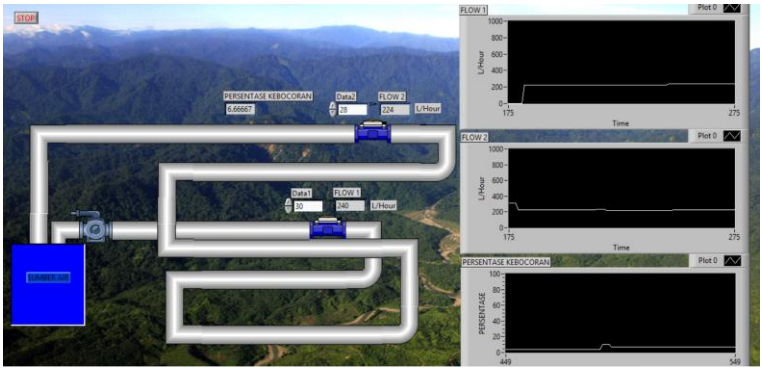
Untuk pengujian kebocoran, yang pertama dilakukan adalah menyambungkan Arduino Uno, *Ethernet shield*, dan sensor *water flowmeter*. Setelah itu buka file tampilan pada *software LabVIEW*. Lalu nyalakan pompa air dan buka *stop* kran pada *prototype* simulasi distribusi air. Amati tampilan pada *LabVIEW* (Gambar 4.7). Setelah itu buka kran air ke-1. Amati tampilan pada *LabVIEW* (Gambar 4.8). tutup kran air ke-1 lalu buka kran air ke-2. Amati tampilan pada *software LabVIEW* (Gambar 4.9). Setelah dilakukan membuka kran air satu per satu, selanjutnya kedua kran dibuka. Amati tampilan pada *LabVIEW* (Gambar 4.10).



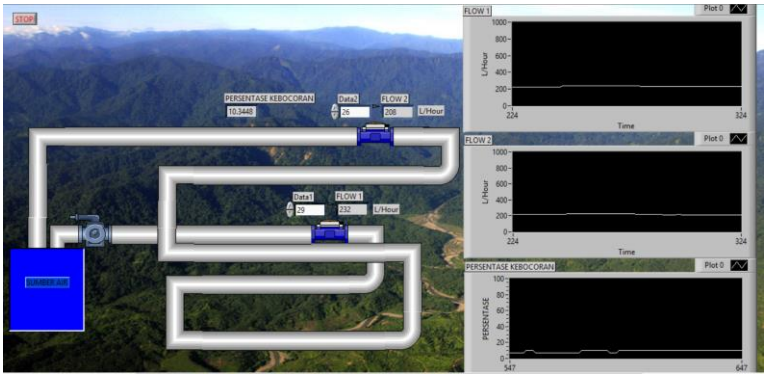
Gambar 4.7. Tampilan *LabVIEW* Pada Saat Keadaan Normal



Gambar 4.8. Tampilan *LabVIEW* Pada Saat Kran Air Ke-1 Dibuka



Gambar 4.9. Tampilan *LabVIEW* Pada Saat Kran Air Ke-2 Dibuka



Gambar 4.10. Tampilan *LabVIEW* Saat Kedua Kran Air Dibuka

Tabel 4.5. Data Pengujian Keseluruhan Alat

Kondisi	FLOW 1 (L/hour)	Data1 (bit)	FLOW 2 (L/hour)	Data2 (bit)	Bocor (%)
Normal	224	28	224	28	0
Kran 1 buka	224	28	216	27	3.57
Kran 2 buka	240	30	224	28	6.67
Keduanya dibuka	232	29	208	26	6.9

Dari gambar-gambar diatas didapatkan data (Tabel 4.5) yakni jika pada saat keadaan normal, hasil FLOW 1 dan FLOW 2 bernilai sama, yaitu 224 L/hour. Hasil pembacaan sensor *water flowmeter* sama karena tidak ada pengurangan debit air. Sehingga kebocoran yang terdeteksi 0%. Sedangkan ketika kran air ke-1 dibuka, hasil FLOW 1 bernilai 224 L/hour dan FLOW 2 bernilai 216 L/hour. Hal ini dikarenakan kran air terbuka, maka debit air berkurang dan juga tekanannya berkurang. Sehingga didapatkan hasil persentase kebocoran 3,57%. Lalu ketika kran air ke-1 ditutup dan yang ke-2 dibuka, hasil dari pembacaan sensor FLOW 1 bernilai 240 L/hour dan FLOW 2 bernilai 224 L/hour. Berbeda pada saat kran air ke-1 dibuka, hasil yang didapatkan pada saat kran air ke-2 dibuka selisih kedua sensor lebih banyak. Hal ini dapat disebabkan oleh beberapa faktor yakni, cara membuka kran air, pembacaan sensor kurang stabil, maupun dari tekanan pada pompa air yang digunakan. Sehingga kebocoran air terdeteksi 6,67%. Pengujian yang terakhir yaitu kedua kran air dibuka. Pada saat kedua kran air terbuka, didapatkan hasil yang jauh berbeda yaitu FLOW 1 232 L/hour dan FLOW 2 208 L/hour. Tentu saja berbeda jauh karena pengurangan debit dan tekanan air lebih banyak. Sehingga didapatkan persentase kebocoran 6,9%

BAB V
PENUTUP

Setelah melakukan perancangan dan pembuatan alat serta pengujian dan analisa, maka dapat ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan untuk pengembangan Tugas Akhir ini.

5.1. Kesimpulan

Dari seluruh tahapan yang sudah dilaksanakan pada penyusunan Tugas Akhir ini, mulai dari studi literatur, perancangan dan pembuatan sampai pada pengujiannya maka dapat disimpulkan bahwa :

1. Pembacaan pada setiap sensor *water flowmeter* berbeda, jadi harus dilakukan kalibrasi pada setiap sensor.
2. Pada *prototype* simulasi distribusi air, ketika kran air salah satu dibuka sedikit, tidak dapat mengetahui persentase kebocoran air.
3. Satuan yang digunakan untuk pembacaan sensor *water flowmeter* terlalu lama, sehingga data yang didapatkan besar.
4. Jika kedua kran air dibuka, didapatkan nilai selisih antara aliran air masuk dan aliran air keluar sebesar 24 L/hour dengan persentase 6,9%.

5.2. Saran

Untuk lebih memperbaiki dan menyempurnakan kinerja dari alat ini, maka perlu disarankan :

1. Pertambahan indikator pada bagian tampilan pada *software LabVIEW* agar dapat langsung mengetahui kebocoran pipa.
2. Perpanjangan jalur instalasi pipa agar lebih akurat untuk mendeteksi kebocoran.
3. Penyesuaian komponen yang digunakan.
4. Dan pengontrolan aliran air agar ketika terjadi kebocoran yang melebihi toleransi, air yang mengalir semakin kecil.

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

DAFTAR PUSTAKA

[1] Ferli S. Irwansyah, Susan Susanti, Neneng Windayani, "Pengolahan Air Sungai Menjadi Air Layak Konsumsi Menggunakan Kulit Pisang Raja Bulu (Musa paradica)", **Jurnal Penelitian**, Program Studi Pendidikan Kimia, UIN Sunan Gunung Djati, Bandung, 2017.

[2] Rizka Masruuro, Dany Setyawan, "Rancang Bangun Kontroler Baterai Charger Untuk Solar Cell", **Tugas Akhir**, Jurusan D3 Teknik Elektro, Fakultas Teknologi Industry, Institut Teknologi Sepuluh Nopember, Surabaya, 2015.

[3] Rafi Maizar, "Flowmeter Air Digital Dengan Toleransi Error Maksimum ± 5%", **Tugas Akhir**, Jurusan Teknik Elektronika, Politeknik Negeri Batam, Batam, 2014.

[4] Duwi Haryanto, "Perancangan Alat Deteksi Letak Kebocoran Pipa Pvc", **Skripsi**, Jurusan Fisika, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Lampung, Lampung, 2016.

[5] Stevani Agnesia Sigiro, Takdir Tamba, Mester Sitepu, Andi Setiono, "Instrumentasi Virtual Menggunakan Labview Dan Soundcard", **Jurnal Penelitian**, Jurusan Fisika, Fakultas Matematika Ilmu Alam, Universitas Sumatra Utara, Medan, 2010.

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

LAMPIRAN A
PROGRAM

A.1 Program *Ethernet* 1

```
#include <SPI.h>
#include <Ethernet.h>
#include <Modbus.h>
#include <ModbusIP.h>

const int sencalc = 100;

volatile int NbTopsFan;
int Calc;
int hallsensor = 2;


void rpm ()
{
  NbTopsFan++;
}

ModbusIP mb;

long ts;

void setup()
{
   byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
   byte ip[] = { 192, 168, 0, 200 };
   mb.config(mac, ip);
   pinMode(hallsensor, INPUT);
   Serial.begin(9600);
   attachInterrupt(0, rpm, RISING);
   mb.addIreg(sencalc);
   NbTopsFan = 0;
   sei();
```

```
}

void loop ()
{
  mb.task();
  if (millis() > ts + 1000) {
      ts = millis();
      cli();
      Calc = (NbTopsFan * 60 / 7.5);
      mb.Ireg(sencalc, Calc);
      Serial.print (NbTopsFan, DEC);
      Serial.print (" L/hour\r\n");
      NbTopsFan = 0;
      sei();
  }
}
```

A.2 Program *Ethernet* 2

```
#include <SPI.h>
#include <Ethernet.h>
#include <Modbus.h>
#include <ModbusIP.h>

const int sencalc = 100;

volatile int NbTopsFan;
int Calc;
int hallsensor = 2;


void rpm ()
{
  NbTopsFan++;
}
```

```
ModbusIP mb;

long ts;
{
   byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xEF };
   byte ip[] = { 192, 168, 0, 202 };
   mb.config(mac, ip);

   pinMode(hallsensor, INPUT);
   Serial.begin(9600);
   attachInterrupt(0, rpm, RISING);
   mb.addIreg(sencalc);
   NbTopsFan = 0;
   sei();

}

void loop ()
{
 mb.task();
 if (millis() > ts + 1000) {
     ts = millis();
     cli();
     Calc = (NbTopsFan * 60 / 7.5);
     mb.Ireg(sencalc, NbTopsFan);
     Serial.print (Calc, DEC);
     Serial.print (" L/hour\r\n");
     NbTopsFan = 0;
     sei();
 }
}
```

A.3 Program *Water Flowmeter Sensor*

```
#include <SPI.h>
#include <Ethernet.h>
```

```
const int sencalc = 100;

volatile int NbTopsFan;
int Calc;
int hallsensor = 2;


void rpm ()
{
  NbTopsFan++;
}

void loop ()
{
  if (millis() > ts + 1000)
    {
       ts = millis();
       cli();
       Calc = (NbTopsFan * 60 / 7.5);
       mb.Ireg(sencalc, NbTopsFan);
       Serial.print (Calc, DEC);
       Serial.print (" L/hour\r\n");
       NbTopsFan = 0;
       sei();
  }
}
```

LAMPIRAN B
DATASHEET

B.1 *Water Flowmeter* Sensor G3/4”

Water Flow Sensor YF-B6

SKU 114991176

Description

Water flow sensor consists of a copper body, a water rotor, and a hall-effect sensor. When water flows through the rotor, rotor rolls. Its speed changes with different rate of flow. The hall-effect sensor outputs the corresponding pulse signal. This one is suitable to detect flow in water dispenser or coffee machine.

Life is longer than plastic body.

Features

•Compact, Easy to Install

•High Sealing Performance

•High Quality Hall Effect Sensor

•RoHS Compliant

Specifications

•Mini. Wokring Voltage: DC 4.5V

•Max. Working Current: 15mA (DC 5V)

•Working Voltage: DC 5V~15V

•Flow Rate Range: 1~30L/min

•Frequency: F=6.6*Q(Q=L/MIN)

•Load Capacity: ≤10mA (DC 5V)

•Operating Temperature: ≤80℃

•Liquid Temperature: ≤120℃

•Operating Humidity: 35%～90%RH

•Water Pressure: ≤1.75MPa

•Storage Temperature: -25～+ 80℃

•Storage Humidity: 25%～95%RH

Part List

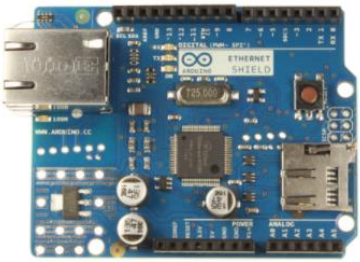
1 x YF-B6 water flow sen

Technical Details
Weight G.W 152g
Battery Exclude

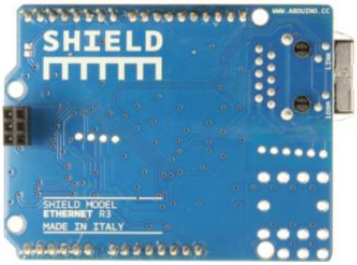
https://www.seeedstudio.com/Water-Flow-Sensor-YF-B6-p-2883.html 6-26-17

B.2 Arduino *Ethernet Shield* V1

Arduino Ethernet Shield V1



Arduino Ethernet Shield R3 Front



Arduino Ethernet Shield R3 Back

Overview

NOTE: this product is currently retired and the documentation will not be kept up-to-date

The Arduino Ethernet Shield V1 connects your Arduino to the internet in mere minutes. Just plug this module onto your Arduino board, connect it to your network with an RJ45 cable (not included) and follow a few simple instructions to start controlling your world through the internet. As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Hundreds of thousands of Arduino boards are already fueling people's creativity all over the world, everyday. Join us now, Arduino is you!

- Requires an Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Ethernet Controller: W5100 with internal 16K buffer
- Connection speed: 10/100Mb
- Connection with Arduino on SPI port

Description

The Arduino Ethernet Shield V1 allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The most recent revision of the board exposes the 1.0 pinout on rev 3 of the Arduino UNO board.

The Ethernet Shield V1 has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with all the Arduino/Genuino boards. The on-board micro SD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.

The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.

Download: arduino-ethernet-shield-06-schematic.pdf, arduino-ethernet-shield-06-reference-design.zip

The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

- IEEE802.3af compliant
- Low output ripple and noise (100mVpp)
- Input voltage range 36V to 57V
- Overload and short-circuit protection
- 9V Output
- High efficiency DC/DC converter: typ 75% @ 50% load
- 1500V isolation (input to output)

NB: the Power over Ethernet module is proprietary hardware not made by Arduino, it is a third party accessory. For more information, see the datasheet

The shield does not come with the PoE module built in, it is a separate component that must be added on.

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 10, 11, 12, and 13 on the Uno and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general I/O. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

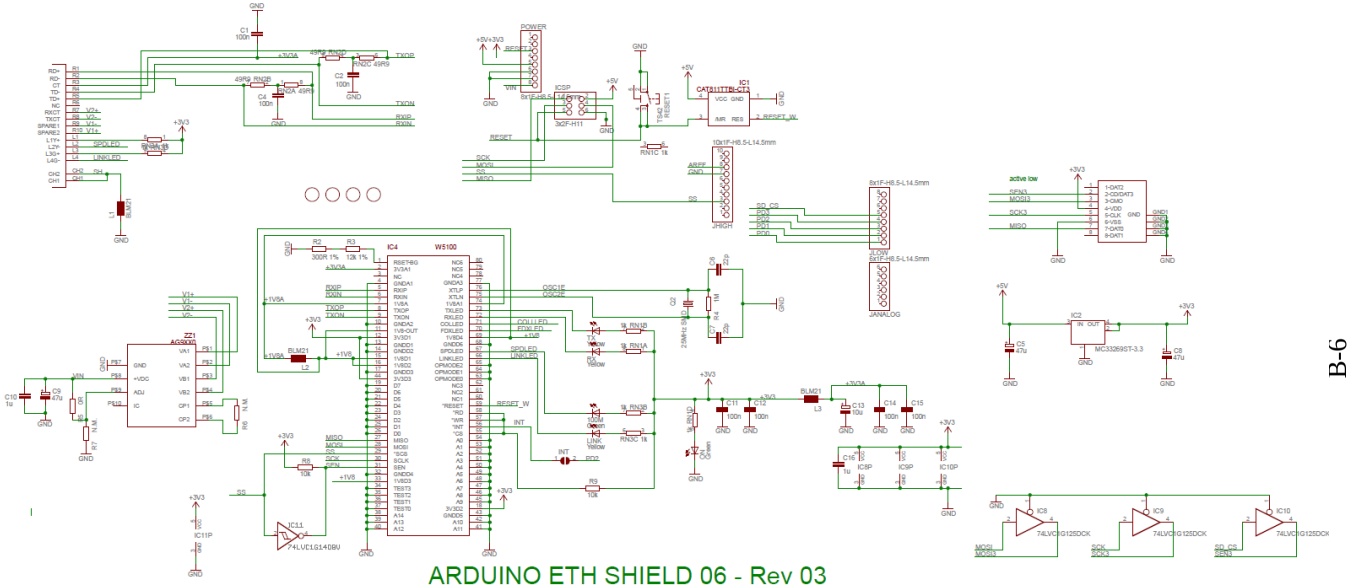
The shield provides a standard RJ45 ethernet jack.

The reset button on the shield resets both the W5100 and the Arduino board.

The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.



ARDUINO ETH SHIELD 06 - Rev 03

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE
Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.

B.2.1 Wiznet W5100

WIZnet

W5100 Datasheet

The W5100 is a full-featured, single-chip Internet-enabled 10/100 Ethernet controller designed for embedded applications where ease of integration, stability, performance, area and system cost control are required. The W5100 has been designed to facilitate easy implementation of Internet connectivity without OS. The W5100 is IEEE 802.3 10BASE-T and 802.3u 100BASE-TX compliant.

The W5100 includes fully hardwired, market-proven TCP/IP stack and integrated Ethernet MAC & PHY. Hardwired TCP/IP stack supports TCP, UDP, IPv4, ICMP, ARP, IGMP and PPPoE which has been proven in various applications for several years. 16Kbytes internal buffer is included for data transmission. No need of consideration for handling Ethernet Controller, but simple socket programming is required.

For easy integration, three different interfaces like memory access way, called direct, indirect bus and SPI, are supported on the MCU side.

Target Applications

The W5100 is well suited for many embedded applications, including:

- Home Network Devices: Set-Top Boxes, PVRs, Digital Media Adapters
- Serial-to-Ethernet: Access Controls, LED displays, Wireless AP relays, etc.
- Parallel-to-Ethernet: POS / Mini Printers, Copiers
- USB-to-Ethernet: Storage Devices, Network Printers
- GPIO-to-Ethernet: Home Network Sensors
- Security Systems: DVRs, Network Cameras, Kiosks
- Factory and Building Automations
- Medical Monitoring Equipments
- Embedded Servers

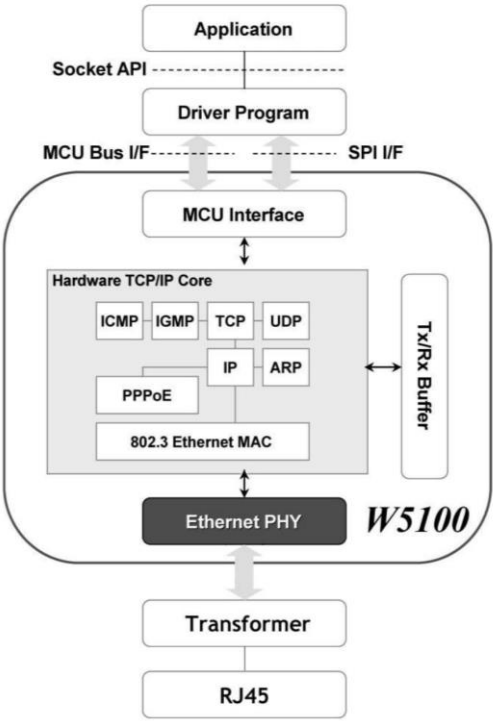
© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Features

- Support Hardwired TCP/IP Protocols : TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet
- 10BaseT/100BaseTX Ethernet PHY embedded
- Support Auto Negotiation (Full-duplex and half duplex)
- Support Auto MDI/MDIX
- Support ADSL connection (with support PPPoE Protocol with PAP/CHAP Authentication mode)
- Supports 4 independent sockets simultaneously
- Not support IP Fragmentation
- Internal 16Kbytes Memory for Tx/Rx Buffers
- 0.18 μm CMOS technology
- 3.3V operation with 5V I/O signal tolerance
- Small 80 Pin LQFP Package
- Lead-Free Package
- Support Serial Peripheral Interface(SPI MODE 0, 3)
- Multi-function LED outputs (TX, RX, Full/Half duplex, Collision, Link, Speed)

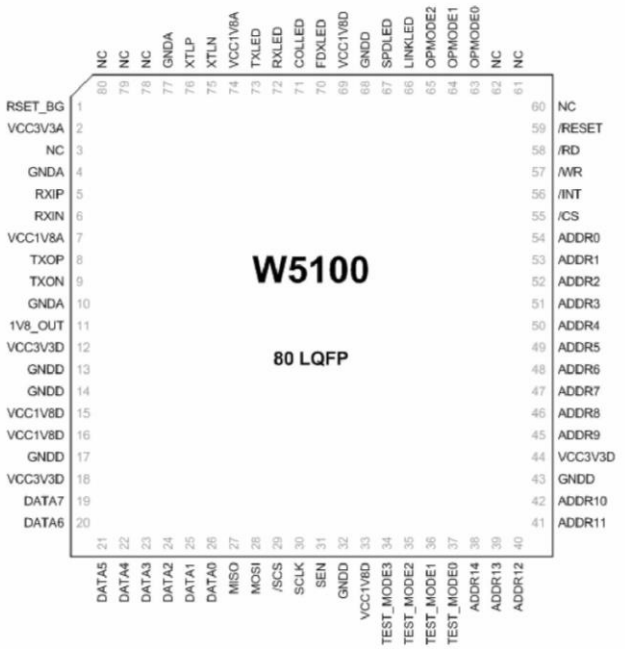
© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Block Diagram



© Copyright 2008 WIZnet Co., Inc. All rights reserved.

1. Pin Assignment



Pinout W5100

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

7. Electrical Specifications

Absolute Maximum Ratings

Symbol	Parameter	Rating	Unit
V_{DD}	DC Supply voltage	-0.5 to 3.6	V
V_{IN}	DC input voltage	-0.5 to 5.5 (5V tolerant)	V
I_{IN}	DC input current	±5	mA
T_{OP}	Operating temperature	-40 to 85	°C
T_{STG}	Storage temperature	-55 to 125	°C

***COMMENT**: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage.

DC Characteristics

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
V_{DD}	DC Supply voltage	Junction temperature is from -55°C to 125°C	3.0		3.6	V
V_{IH}	High level input voltage		2.0		5.5	V
V_{IL}	Low level input voltage		- 0.5		0.8	V
V_{OH}	High level output voltage	I_{OH} = 2, 4, 8, 12, 16, 24 mA	2.0		3.6	V
V_{OL}	Low level output voltage	I_{OL} = -2, -4, -8, -12, -16, -24 mA	0.0		0.4	V
I_I	Input Current	$V_{IN} = V_{DD}$			±5	μA

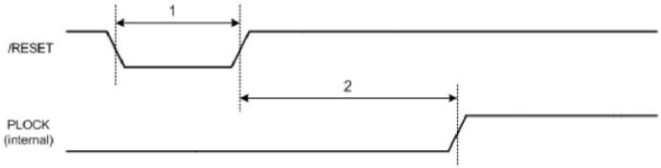
POWER DISSIPATION

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
P_{10Base}	Power consumption in 10BaseT	Vcc 3.3V Temperature 25°C	-	138	183	mA
$P_{100Base}$	Power consumption in 100BaseT	Vcc 3.3V Temperature 25°C	-	146	183	mA

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

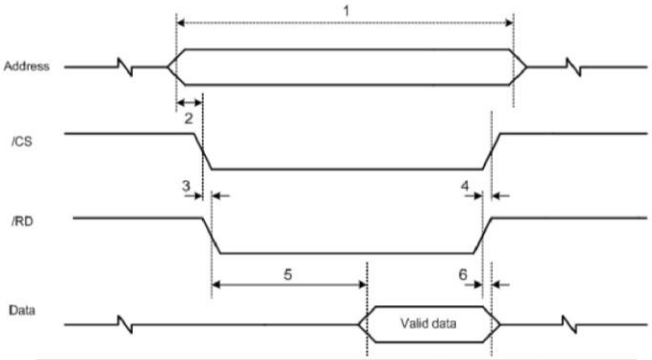
AC Characteristics

Reset Timing



	Description	Min	Max
1	Reset Cycle Time	2 us	-
2	/RESET to internal PLOCK	-	10 ms

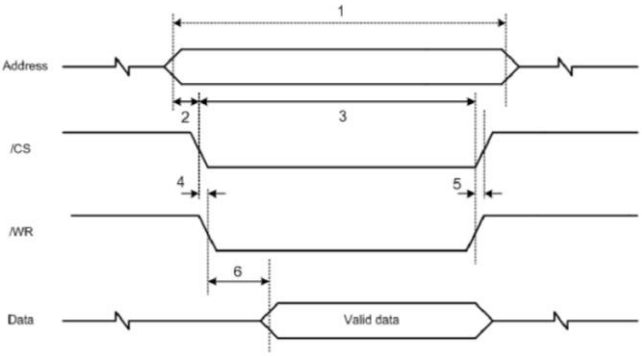
Register/Memory READ Timing



	Description	Min	Max
1	Read Cycle Time	80 ns	-
2	Valid Address to /CS low time	8 ns	-
3	/CS low to /RD low time	-	1 ns
4	/RD high to /CS high time	-	1 ns
5	/RD low to Valid Data Output time	-	80 ns
6	/RD high to Data High-Z Output time	-	1 ns

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

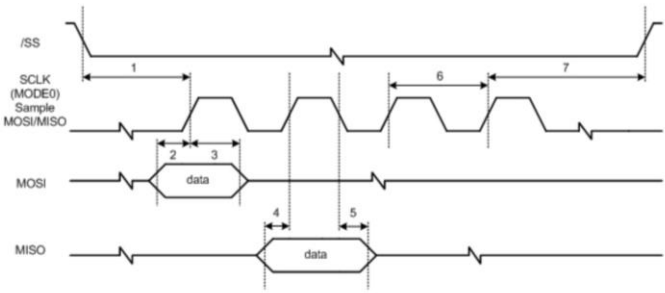
Register/Memory WRITE Timing



	Description	Min	Max
1	Write Cycle Time	70 ns	-
2	Valid Address to /CS low time	7 ns	-
3	/CS low to /WR high time	70 ns	-
4	/CS low to /WR low time	-	1 ns
5	/WR high to /CS high time	-	1 ns
6	/WR low to Valid Data time	-	14 ns

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

SPI Timing



	Description	Mode	Min	Max
1	/SS low to SCLK high	Slave	21 ns	-
2	Input setup time	Slave	7 ns	-
3	Input hold time	Slave	28 ns	-
4	Output setup time	Slave	7 ns	14 ns
5	Output hold time	Slave	21 ns	-
6	SCLK time	Slave	70 ns	
7	SCLK high to /SS high	Slave	21ns	

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Crystal Characteristics

Parameter	Range
Frequency	25 MHz
Frequency Tolerance (at 25℃)	±30 ppm
Shunt Capacitance	7pF Max
Drive Level	100uW
Load Capacitance	27pF
Aging (at 25℃)	±3ppm / year Max

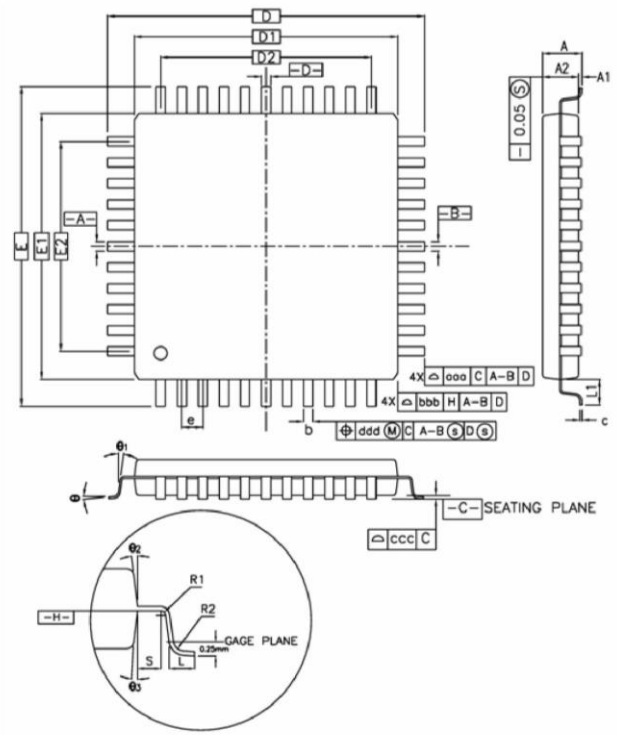
Transformer Characteristics

Parameter	Transmit End	Receive End
Turn Ratio	1:1	1:1
Inductance	350 uH	350 uH

Symmetrical TX & RX channels for auto MDI/MDIX capability

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Package Descriptions



Above diagram shows PIN dimension. All 80 pins are not displayed.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	-	-	1.60	-	-	0.063
A1	0.05	-	0.15	0.002	-	0.006

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

A2	1.35	1.40	1.45	0.053	0.055	0.057
D		12.00 BSC.			0.472 BSC.	
D1		10.00 BSC.			0.393 BSC.	
E		12.00 BSC.			0.472 BSC.	
E1		10.00 BSC.			0.393 BSC.	
R2	0.08	-	0.20	0.003	-	0.008
R1	0.08	-	-	0.003	-	-
θ	0°	3.5°	7°	0°	3.5°	7°
θ_1	0°	-	-	0°	-	-
θ_2	11°	12°	13°	11°	12°	13°
θ_3	11°	12°	13°	11°	12°	13°
c	0.09	-	0.20	0.004	-	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00 REF			0.039 REF	
S	0.20	-	-	0.008	-	-
b	0.13	0.16	0.23	0.005	0.006	0.009
e		0.40 BSC			0.016 BSC	
D2		7.60			0.299	
E2		7.60			0.299	
aaa		0.20			0.008	
bbb		0.20			0.008	
ccc		0.08			0.003	
ddd		0.07			0.003	

Note :

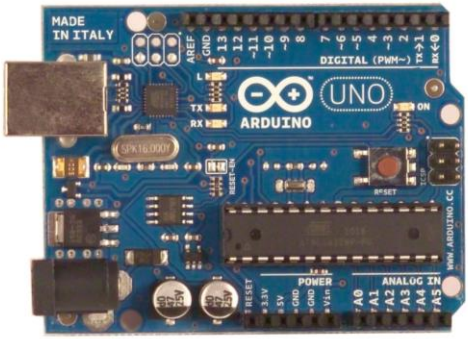
1. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.

2. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.
ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.
DAMBAR CAN NOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND AN ADJACENT LEAD IS 0.07mm FOR 0.4mm AND 0.5mm PITCH PACKAGES.

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

B.3 Arduino Uno R3

Arduino UNO



CE

Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Index

Technical Specifications — Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials — Page 6

Terms & Conditions — Page 7

Enviromental Policies
half sqm of green via Impatto Zero® — Page 7

radiospares RADIONICS

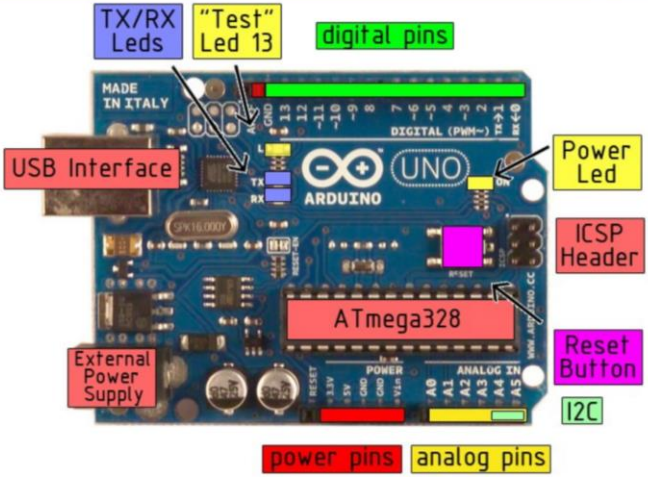
Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



RS radiospares RADIONICS ALLIED ELECTRONICS

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno 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. TThese pins are connected to the corresponding pins of the ATmega8U2 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.

RS radiospares RADIONICS ALLIED ELECTRONICS

The Uno has 6 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 AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the Wire library.

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 Atmega328 ports.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

RS radiospares RADIONICS ALLIED ELECTRONICS

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

RS radiospares RADIONICS ALLIED ELECTRONICS

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install Windows Install Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

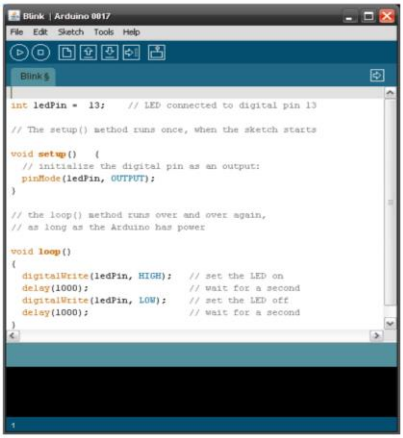
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

Once you have your skecth you'll see something very close to the screenshot on the right.

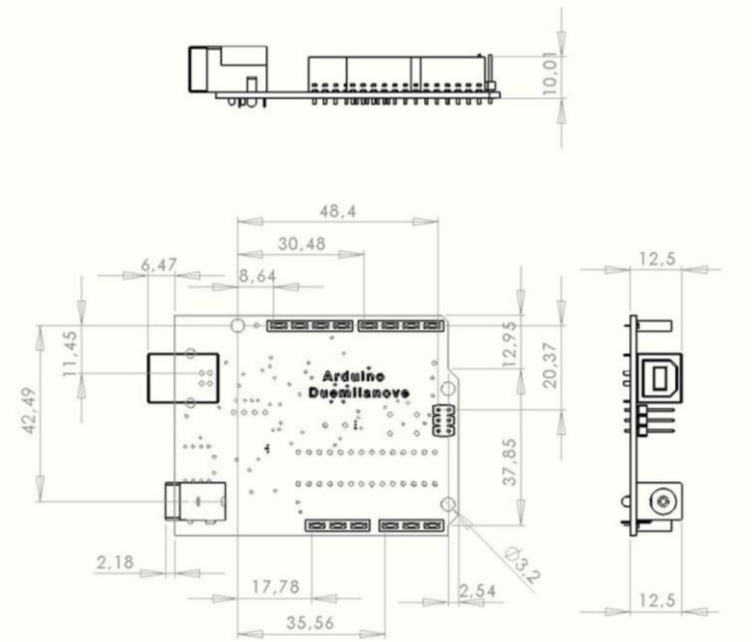
In **Tools>Board** select

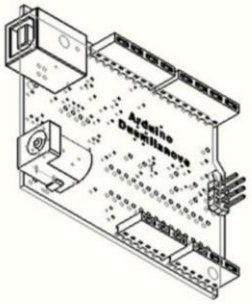
Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



RS radiospares RADIONICS ALLIED ELECTRONICS

Dimensioned Drawing





RS radiospares RADIONICS ALLIED ELECTRONICS

Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

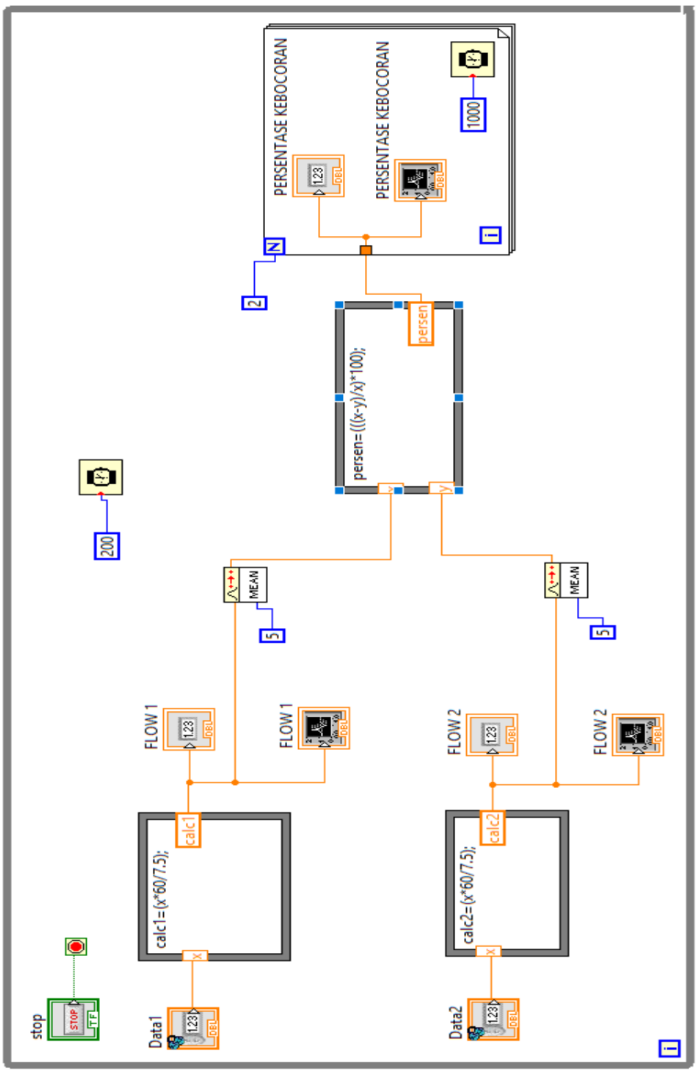
Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

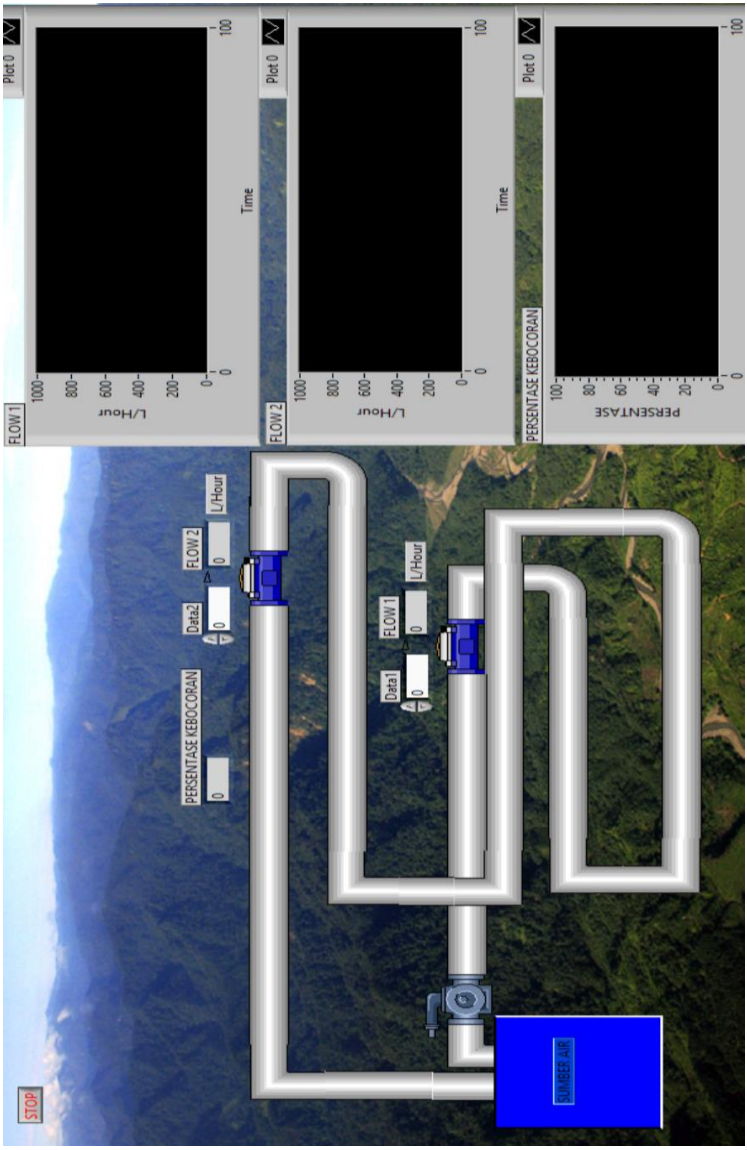
-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C
TAMPILAN *LABVIEW*

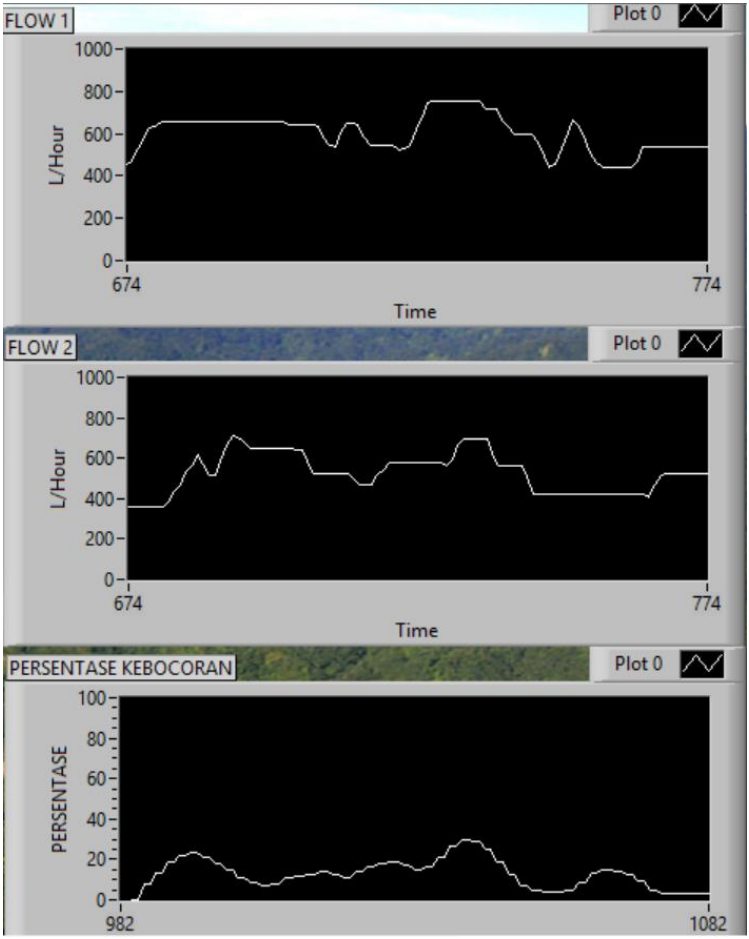
C.1. *Block Diagram*



C.2. *Front Panel*



C.3. Tampilan Grafik



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

DAFTAR RIWAYAT HIDUP

Nama : Farrah Fadilah
TTL : Surabaya, 5 Juni 1996
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Jl. Ampel Rachmad no 7, Surabaya / Perumahan Bukit Bambe AP-11, Gresik.
Telp/Hp : 085748106823
E-mail : farrah.0506@gmail.com

RIWAYAT PENDIDIKAN

1. 2002 – 2008 : SD Negeri Bambe I
2. 2008 – 2011 : SMP Negeri 16 Surabaya
3. 2011 – 2014 : SMA Muhammadiyah 4 Surabaya
4. 2014 – 2018 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. DELTA JAYA MAS (DJM)
2. Kerja Praktek di PT. PLN (Persero) Area Surabaya Barat

PENGALAMAN ORGANISASI

1. Staff Departemen Hubungan Luar Periode 2015/2016 Himpunan Mahasiswa D3 Teknik Elektro, Institut Teknologi Sepuluh Nopember
2. Kepala Biro Devisi Pengembangan, Devisi Big Event Periode 2016/2017 Himpunan Mahasiswa D3 Teknik Elektro, Institut Teknologi Sepuluh Nopember

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