



PROYEK AKHIR - VE180626

KONTROL KECEPATAN MOTOR DC DENGAN BEBAN BANDUL

Sofyan A Rahman
NRP 10311500000094

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

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



FINAL PROJECT - VE180626

***MOTOR DC SPEED CONTROL WITH PENDULUM
LOAD***

Sofyan A Rahman
NRP 10311500000094

Supervisor
Ir. Josaphat Pramudijanto, M.Eng

*Electrical and Automation Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2019*

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

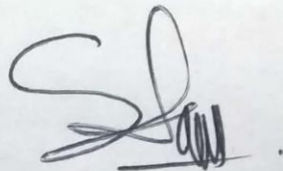
PERNYATAAN KEASLIAN PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul **“KONTROL KECEPATAN MOTOR DC DENGAN BEBAN BANDUL”** 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, Januari 2019



Sofyan A Rahman
NRP 10311500000094

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

KONTROL KECEPATAN MOTOR DC DENGAN BEBAN BANDUL

PROYEK AKHIR

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

Menyutujui :

Dosen Pembimbing



Ir. Josaphat Pramudijanto, M.Eng.

NIP. 1962005 199003 1 003

SURAABAYA
JANUARI, 2019

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

KONTROL KECEPATAN MOTOR DC DENGAN BEBAN BANDUL

Nama Mahasiswa : Sofyan A Rahman
NRP : 10311500000094
Pembimbing I : Ir. Josaphat Pramudijanto, M.Eng.
NIP : 19621005 199003 1 003

ABSTRAK

Kerusakan alat yang diakibatkan oleh getaran dalam proses pemakaian dan faktor lingkungan sering terjadi dikarenakan kurangnya pengujian awal terhadap batas getaran yang mampu diterima oleh alat tersebut. Alat uji getaran secara fisik masih jarang dijual dipasaran, maka dari itu dibuatlah alat untuk uji getaran agar bisa menguji alat yang akan digunakan.

Proyek Akhir ini membahas tentang kontrol kecepatan motor DC menggunakan kontrol PID pada alat uji getaran. Cara kerja alat adalah dengan memanipulasi getaran yang dihasilkan motor 24 VDC dengan cara mengatur kecepatan motor 24 VDC tersebut menggunakan kontrol PID yang mengambil data dari sensor *Optocoupler* TCST 1202. Hasil tersebut akan di tampilkan pada LCD (*liquid crystal display*) dengan satuan RPM.

Getaran yang dihasilkan pada alat ini yaitu 987Hz dengan kecepatan maksimal motor DC yakni 2322rpm. Alat ini menggunakan kontrol PID untuk mengatur kecepatan motor dengan nilai K_p sebesar 2. Dengan spesifikasi demikian alat ini dapat digunakan untuk menguji ketahanan suatu benda terhadap getaran

Kata Kunci : Arduino Uno, Getaran, *Optocoupler* TCST 1202, Kontrol PID

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

MOTOR DC SPEED CONTROL WITH PENDULUM LOAD

Student's Name : Sofyan A Rahman
Registration Number : 10311500000094
Supervisor I : Ir. Josaphat Pramudijanto, M.Eng.
ID Number : 19621005 199003 1 003

ABSTRACT

Damage to the equipment caused by vibrations in the process of use and environmental factors often occurs due to the lack of initial testing of the vibration limits that can be accepted by the tool. Physical vibration testing equipment is rarely sold in the market, so a tool for vibration testing is made so that it can test the tool to be used.

This Final Project discusses the control of DC motor speed using PID control on vibration test equipment. The way the tool works is by manipulating the vibrations produced by the 24 VDC motor by adjusting the 24 VDC motor speed using a PID control that takes data from the TCST 1202 Optocoupler sensor. The results will be displayed on LCD (liquid crystal display) with RPM units.

The vibration produced on this device is 987Hz with a maximum DC motor speed of 2322rpm. This tool uses a PID control to adjust the motor speed with a Kp value of 2. With such specifications this tool can be used to test the resistance of an object to vibration

Keywords: Arduino Uno, Vibration, TCST 1202 Optocoupler Sensor, PID Control

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

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Proyek 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.

Proyek Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

KONTROL KECEPATAN MOTOR DC DENGAN BEBAN BANDUL

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

1. Kedua orangtua yang selalu memberi dukungan, doa, dan memberi perhatian kepada penulis.
2. Bapak Ir. Joko Susila, MT selaku Ketua Departemen Teknik Elektro Otomasi, FV-ITS.
3. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku dosen pembimbing
4. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

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

Surabaya, 22 Januari 2019

Sofyan A Rahman

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

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Batasan Masalah	1
1.4 Tujuan	2
1.5 Sistematika Laporan	2
1.6 Relevansi	3
BAB II TEORI PENUNJANG	5
2.1 Getaran	5
2.2 <i>Optocoupler</i> Sensor TCST 1202	6
2.3 Kontrol PID	7
2.4 LCD (<i>Liquid Crystal Display</i>)	9
2.5 Arduino	9
2.5.1 Arduino UNO	9
2.5.2 IDE Arduino	10
2.6 <i>Power Supply Adaptor Switching</i>	11
2.7 <i>Driver Motor</i> BTS7960 43A	12
2.8 <i>Spring</i> Tekan	13
2.9 <i>Potensiometer</i>	14
2.10 Motor DC 24 Volt	14
2.11 Kontrol Kecepatan pada Motor DC	15
BAB III RANCANGAN DAN PEMBUATAN ALAT	19
3.1 Blok Fungsional Sistem	19
3.2 Perancangan Perangkat Keras (<i>Hardware</i>)	20
3.2.1 Perancangan Alat Uji Getaran	20
3.2.2 Rangkaian Alat Uji Getaran	22
3.2.3 Rangkaian <i>Optocoupler</i> Sensor	24
3.2.4 Rangkaian LCD (<i>Liquid Crystal Display</i>)	24
3.2.5 Rangkaian <i>Driver Motor</i> BTS7960 43A	24
3.3 Perancangan Perangkat Lunak (<i>Software</i>)	25

3.3.1	Pemrograman Pembacaan Sensor <i>Optocoupler</i>	25
3.4	Pejelasan Kontrol Kecepatan Motor	26
BAB IV	PENGUJIAN DAN ANALISA	29
4.1	Pengujian LCD 16x2 I2C.....	29
4.2	Pengujian Sensor <i>Optocoupler</i> TCST1202	30
4.3	Pengujian Motor DC dengan <i>Open Loop Control</i>	32
4.4	Pengujian Karakteristik Motor DC	33
4.5	Pengujian Motor DC dengan <i>Close Loop Control</i>	33
4.6	Pengujian Kecepatan Motor Terhadap Getaran.	35
BAB V	PENUTUP	37
5.1	Kesimpulan	37
5.2	Saran	37
DAFTAR PUSTAKA		39
LAMPIRAN A		A-1
Alat Uji Getaran.....		A-1
Program Kontrol PID.....		A-2
Program FFT Getaran		A-6
LAMPIRAN B		B-1
<i>Datasheet</i> Arduino UNO		B-1
<i>Datasheet</i> BTS7960.....		B-9
DAFTAR RIWAYAT HIDUP		C-1

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Getaran Periodik Sederhana (Gerak Harmonik)	6
Gambar 2.2 Sensor <i>Optocoupler TCST 1202</i>	6
Gambar 2.3 Grafik Kontrol PID	8
Gambar 2.4 LCD (<i>Liquid Crystal Display</i>)	9
Gambar 2.5 Arduino Uno R3.....	10
Gambar 2.6 IDE Arduino	11
Gambar 2.7 <i>Power Supply Adaptor Switching</i>	12
Gambar 2.8 <i>Driver Motor BTS7960 43A</i>	13
Gambar 2.9 <i>Spring Tekan</i>	13
Gambar 2.10 <i>Potensiometer 1K Ohm</i>	14
Gambar 2.11 Motor DC 24 Volt	15
Gambar 2.12 Siklus Sinyal PWM pada Arduino.	17
Gambar 3.1 Blok Diagram Fungsional Alat Uji Getaran	19
Gambar 3.2 Perancangan Alat Uji Getaran	21
Gambar 3.3 Rangkaian Alat Uji Getaran.....	22
Gambar 3.4 Rangkaian <i>Optocoupler</i> Sensor ke Arduino Uno	23
Gambar 3.5 Rangkaian LCD 16x2 I2C ke Arduino Uno.....	23
Gambar 3.6 Rangkaian <i>Driver motor BTS7960 43A</i>	24
Gambar 3.7 <i>Flowchart</i> Pembacaan Sensor <i>Optocoupler</i>	25
Gambar 3.8 Rangkaian Kontrol Kecepatan Motor DC	26
Gambar 4.1 Foto Keseluruhan Alat	29
Gambar 4.2 Program Tampilan LCD 16x2 I2C	30
Gambar 4.3 Tampilan LCD 16x2 I2C	30
Gambar 4.4 Penempatan Sensor <i>Optocoupler TCST1202</i>	31
Gambar 4.5 Kontrol PID dengan Kp:1 , Ki:0 , Kd:0	34
Gambar 4.6 Kontrol PID dengan Kp:2 , Ki:0 , Kd:0	34

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

DAFTAR TABEL

	HALAMAN
Tabel 4.1	Kalibrasi Sensor <i>Optocoupler</i>31
Tabel 4.2	Kecepatan Motor Dc dengan <i>Open Loop Control</i>32
Tabel 4.3	Pengujian Karakteristik Motor DC33
Tabel 4.4	Kecepatan Motor Terhadap Getaran35

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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Getaran adalah salah satu faktor yang mempengaruhi ketahanan suatu barang. Karena beberapa barang yang peka terhadap getaran akan rusak atau tidak bisa digunakan lagi saat diberikan getaran yang melampaui ketahanannya. Kebanyakan barang elektronik seperti hardisk, handphone, laptop dan barang barang lainnya mempunyai batas getaran yang bisa diterimanya.

Pengujian barang elektronik terhadap getaran berperan penting dalam penentuan batas maksimal ketahanan barang tersebut terhadap getaran. Kerusakan alat yang diakibatkan oleh getaran dalam proses pemakaian dan faktor lingkungan sering terjadi dikarenakan kurangnya pengetahuan tentang batas maksimal getaran yang mampu diterima alat yang menyebabkan umur alat lebih pendek dari batas umur yang telah ditetapkan.

Hal inilah yang membuat penulis perlu mengembangkan Kontrol Kecepatan Motor DC dengan Beban Bandul pada alat uji getaran untuk menguji daya tahan sebuah komponen terhadap suatu getaran sebelum penggunaan pada kehidupan sehari-hari dan juga untuk *maintenance predict* agar mengetahui batasan maksimal getaran yang bisa diterima sebuah komponen elektronik atau alat yang diuji.

1.2 Permasalahan

Belum terdapat alat yang digunakan sebagai penguji ketahanan suatu alat terhadap getaran menggunakan kontrol PID.

1.3 Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari Proyek Akhir ini adalah :

1. Sensor yang digunakan sebagai pembaca kecepatan motor DC adalah *Optocoupler* TCST1202
2. Kontrol kecepatan motor DC menggunakan kontrol PID
3. *Driver motor* yang digunakan yaitu BTS7960 43A
4. Batas kemampuan kerja beban pada alat uji getaran sebesar 5 kilogram
5. Menggunakan Motor DC 24 Volt dengan kecepatan maksimal 2322 rpm
6. Getaran maksimal yang dapat dihasilkan alat ini yakni 987Hz

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

1.4 Tujuan

Tujuan kami menuliskan Proyek Akhir ini adalah:

1. Merancang alat uji getaran menggunakan kontrol kecepatan motor menggunakan beban bandul pada alat uji getaran
2. Membuat *hardware* menggunakan Arduino dengan kontrol PID untuk mengatur kecepatan motor.
3. Dapat diimplementasikan pada pengujian barang untuk mengetahui ketahanan terhadap getaran.

1.5 Sistematika Laporan

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

Bab I Pendahuluan

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

Bab II Teori Penunjang

Bab ini menjelaskan tentang tinjauan pustaka, getaran, *Optocoupler* sensor TCST 1202, kontrol PID, LCD (*Liquid Crystal Display*), Arduino Uno, *power supply switching*, *Driver motor* BTS 7960, *spring* tekan, *potensiometer*, motor dc 24 Volt.

Bab III Perancangan dan Pembuatan Alat

Bab ini membahas perancangan *hardware* maupun *software* pada Alat Uji Getaran dengan Kontrol PID Menggunakan Arduino 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 Proyek Akhir dan saran – saran untuk pengembangan alat ini lebih lanjut.

1.6 Relevansi

Diharapkan pada Proyek Akhir “Kontrol Kecepatan Motor DC dengan Beban Bandul” nantinya bisa digunakan untuk mengatur kecepatan motor pada alat penguji getaran sehingga pada saat digunakan pengguna dapat mengatur kecepatan motor sesuai dengan yang diinginkan.

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

BAB II

TEORI PENUNJANG

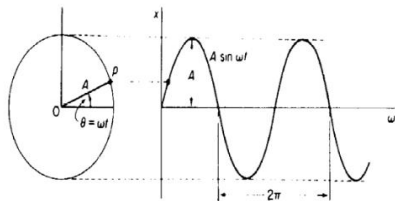
2.1 Getaran [1]

Getaran adalah gerak bolak-balik yang terjadi secara teratur di sekitar titik setimbangnya. Karena terjadi secara teratur. Getaran sering disebut gerak berkala atau gerak periodik. Getaran banyak dimanfaatkan dalam kehidupan sehari-hari. Getaran senar-senar gitar yang dipetik menghasilkan musik yang merdu, getaran bandul dapat dimanfaatkan sebagai penunjuk waktu, dan getaran pada alat suspensi mobil memberi kenyamanan dalam berkendara. Terutama ketika mobil melintasi jalan tidak rata. Tidak semua getaran bermanfaat, Getaran mesin-mesin kadang-kadang tidak menyenangkan. Karena suaranya sangat mengganggu kenyamanan. Getaran mesin mobil dapat mengurangi kenyamanan dalam berkendara. Itulah sebabnya banyak teknologi dikembangkan untuk meredam getaran-getaran yang merugikan.

Secara umum dikenal dua macam jenis getaran berdasarkan proses terjadinya getaran, yakni getaran Bebas, adalah getaran yang terjadi ketika sistem mekanis dimulai dengan adanya gaya awal yang bekerja pada sistem itu sendiri, lalu dibiarkan bergetar secara bebas. Getaran bebas akan menghasilkan frekuensi yang natural karena sifat dinamika dari distribusi massa dan kekuatan yang membuat getaran, contohnya bandul yang ditarik kemudian dilepaskan dan dibiarkan menghasilkan getaran sampai pergerakan bandul tersebut berhenti. Getaran Paksa Getaran Paksa, ialah suatu getaran yang terjadi ketika gerakan bolak-balik karena adanya gaya luar yang secara paksa menciptakan getaran pada sistem. Getaran adalah gerak bolak-balik yang terjadi secara teratur di sekitar titik setimbangnya. Karena terjadi secara teratur. Getaran sering disebut gerak berkala atau gerak periodik.

Getaran banyak dimanfaatkan dalam kehidupan sehari-hari. Getaran senar-senar gitar yang dipetik menghasilkan musik yang merdu, getaran bandul dapat dimanfaatkan sebagai penunjuk waktu, dan getaran pada alat suspensi mobil memberi kenyamanan dalam berkendara. Terutama ketika mobil melintasi jalan tidak rata. Tidak semua getaran bermanfaat, Getaran mesin-mesin kadang-kadang tidak menyenangkan. Karena suaranya sangat mengganggu kenyamanan. Getaran mesin mobil dapat mengurangi kenyamanan dalam berkendara. Itulah sebabnya banyak teknologi dikembangkan untuk meredam getaran-getaran yang merugikan.

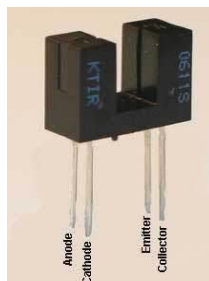
Secara umum dikenal dua macam jenis getaran berdasarkan proses terjadinya getaran, getaran bebas adalah getaran yang terjadi ketika sistem mekanis dimulai dengan adanya gaya awal yang bekerja pada sistem itu sendiri, lalu dibiarkan bergetar secara bebas. Getaran bebas akan menghasilkan frekuensi yang natural karena sifat dinamika dari distribusi massa dan kekuatan yang membuat getaran. seperti bandul yang ditarik kemudian dilepaskan dan dibiarkan menghasilkan getaran sampai pergerakan bandul tersebut berhenti.. Gambar getaran periodik sederhana seperti yang ditunjukkan pada Gambar 2.1.



Gambar 2.1. Getaran Periodik Sederhana (Gerak Harmonik)

2.2 *Optocoupler Sensor TCST 1202* [2]

Optocoupler adalah komponen elektronika yang berfungsi sebagai penghubung berdasarkan cahaya optic. Pada dasarnya *Optocoupler* terdiri dari 2 bagian utama yaitu *transmitter* yang berfungsi sebagai pengirim cahaya optik dan *receiver* yang berfungsi sebagai pendeteksi sumber cahaya. Masing-masing bagian *Optocoupler* (transmitter dan receiver) tidak memiliki hubungan konduktif rangkaian secara langsung. Gambar *Optocoupler* TCST 1202 dapat dilihat pada Gambar 2.2



Gambar 2.2. Sensor *Optocoupler* TCST 1202

2.3 Kontrol PID [3]

PID (*Proportional Integral Derivative Controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Pengontrol PID akan memberikan aksi kepada motor DC berdasarkan besar *error* yang diperoleh. Motor DC akan menjadi aktuator yang mengatur kecepatan motor DC yang diinginkan disebut dengan *Set Point*. *Error* adalah perbedaan dari *Set Point* dengan level air actual. Untuk lebih memaksimalkan kerja pengontrol diperlukan nilai batas minimum dan maksimum yang akan membatasi nilai *Manipulated Variable* yang dihasilkan. Komponen kontrol PID ini terdiri dari tiga jenis yaitu *Proportional*, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant.

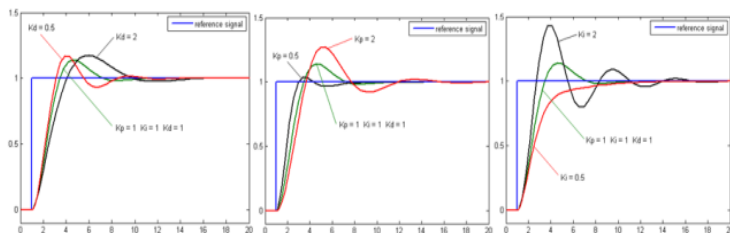
Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika $u = G(s) \cdot e$ maka $u = K_p \cdot e$ dengan K_p adalah Konstanta Proporsional. K_p berlaku sebagai *Gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya *rise time* dan *settling time*. Pengontrol proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya).

Pengontrol Integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol (*Error Steady State* = 0). Jika sebuah pengontrol tidak memiliki unsur integrator, pengontrol proporsional tidak mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol. Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = \int e(t) dt$ dengan K_i adalah konstanta Integral, dan dari persamaan di atas, $G(s)$ dapat dinyatakan sebagai $u = K_i \cdot [\Delta e / \Delta t]$, Jika $e(T)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki *error*. Jika $e(T)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan *output* berosilasi karena menambah orde

sistem. Keluaran pengontrol ini merupakan hasil penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, maka keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Sinyal keluaran pengontrol integral merupakan luas bidang yang dibentuk oleh kurva kesalahan / *error*.

Keluaran pengontrol diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengontrol akan mengakibatkan perubahan yang sangat besar dan cepat. Ketika masukannya tidak mengalami perubahan, keluaran pengontrol juga tidak mengalami perubahan, sedangkan apabila sinyal masukan berubah mendadak dan menaik (berbentuk fungsi *step*), keluaran menghasilkan sinyal berbentuk impuls. Jika sinyal masukan berubah naik secara perlahan (fungsi *ramp*), keluarannya justru merupakan fungsi *step* yang besar magnitudenya sangat dipengaruhi oleh kecepatan naik dari fungsi *ramp* dan factor konstanta K_d . Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s)=s.K_d$. Dari persamaan di atas, nampak bahwa sifat dari kontrol D ini dalam konteks “kecepatan” atau rate dari *error*. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi *error* yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan *error* sehingga saat *error* statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivative tidak dapat dipakai sendiri.

Karakteristik pengontrol PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta K_p , K_i dan K_d akan mengakibatkan penonjolan sifat dari masing-masing elemen.. Adapun beberapa grafik dapat menunjukkan bagaimana respon dari sitem terhadap perubahan K_p , K_i dan K_d seperti pada Gambar 2.3.

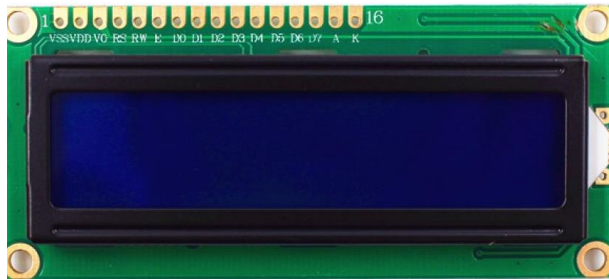


Gambar 2.3. Grafik Kontrol PID

PID *Controller* adalah *controller* yang penting yang sering digunakan dalam industri. Sistem pengendalian menjadi bagian yang tidak bisa terpisahkan dalam proses kehidupan ini khususnya dalam bidang rekayasa industri, karena dengan bantuan sistem pengendalian maka hasil yang diinginkan dapat terwujud. Sistem pengendalian dibutuhkan untuk memperbaiki tanggapan sistem dinamik agar didapat sinyal keluaran seperti yang diinginkan. Sistem kendali yang baik mempunyai tanggapan yang baik terhadap sinyal masukan yang beragam.

2.4 *Liquid Crystal Display* [4]

LCD (*Liquid Crystal Display*) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD yang dipakai merupakan tipe berkarakter 16×2 baris, yang dapat menampilkan 16 karakter dengan 2 baris. Bentuk fisik dari LCD dapat dilihat pada Gambar 2.4



Gambar 2.4 LCD (*Liquid Crystal Display*)

2.5 **Arduino** [5]

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronika dalam berbagai bidang. *Hardware* dalam Arduino memiliki prosesor Atmel AVR dan menggunakan *software* dan bahasa sendiri.

2.5.1 **Arduino Uno** [5]

Perangkat keras dalam Arduino memiliki beberapa jenis, yang mempunyai kelebihan dan kekurangan dalam setiap papannya. Penggunaan jenis Arduino disesuaikan dengan kebutuhan, hal ini yang akan mempengaruhi dari jenis prosesor yang digunakan. Jika semakin kompleks perancangan dan program yang dibuat, maka harus

sesuai pula jenis *controller* yang digunakan. Yang membedakan antara Arduino yang satu dengan yang lainnya adalah penambahan fungsi dalam setiap papan sirkuitnya dan jenis mikrokontroler yang digunakan. Dalam Proyek Akhir perancangan *prototype* jenis Arduino yang digunakan adalah Arduino Uno, Sesuai dengan Gambar 2.5

Papan Arduino Uno dapat mengambil daya dari USB *port* pada komputer dengan menggunakan USB *charger* atau dapat pula mengambil daya dengan menggunakan suatu DC *adapter* dengan tegangan 12 Volt. Jika tidak terdapat *power supply* yang melalui DC *adapter*, maka papan Arduino akan mengambil daya dari USB *port*. Tetapi apabila diberikan daya melalui DC *adapter* secara bersamaan dengan USB *port* maka papan Arduino akan mengambil daya melalui DC *adapter* secara otomatis.



Gambar 2.5 Arduino Uno R3

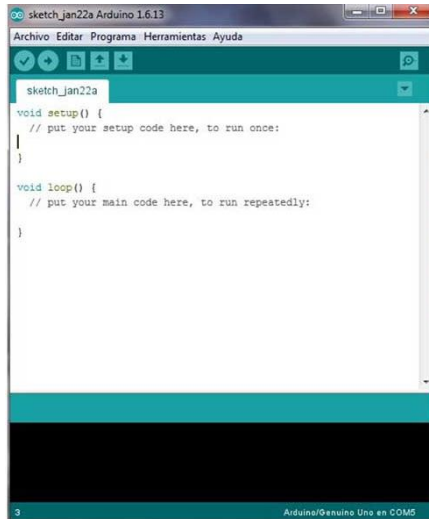
2.5.2 IDE Arduino [5]

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara compiler Arduino dengan mikrokontroler.

Program yang ditulis dengan menggunakan Arduino *Software* (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi *.ino*. Teks editor pada Arduino *Software* memiliki fitur” seperti *cutting/paste* dan

seraching/replacing sehingga memudahkan kamu dalam menulis kode program.

Pada *Software* Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, compile, dan upload program. Di bagian bawah paling kanan *Sotware* Arduino IDE, menunjukan *board* yang terkonfigurasi beserta COM Ports yang digunakan. Gambar tampilan *software* IDE Arduino seperti pada Gambar 2.6



Gambar 2.6 IDE Arduino

2.6 Power Supply Adaptor Switching [6]

Power Supply atau Catu daya ini memerlukan sumber energi listrik yang kemudian mengubahnya menjadi energi listrik yang dibutuhkan oleh perangkat elektronika lainnya. Oleh karena itu, *Power Supply* kadang-kadang disebut juga dengan istilah *Electric Power Converter*.

Power supply 12V 3 A ini memiliki sumber tegangan input sebesar 86-240 V AC dan tegangan output sebesar 12V DC. Daya maksimal yang mampu dihasilkan adalah sebesar 3A (36W). Produk *power supply* ini didesain dengan dimensi P 8,5 x L 5,8 x T 3,4 (CM).



Gambar 2.7 Power Supply Adaptor Switching

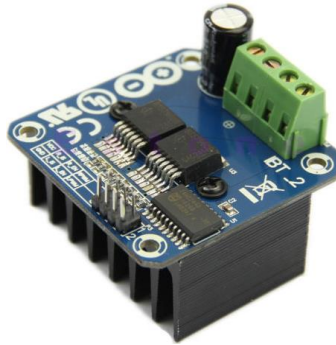
2.7 Driver Motor BTS7960 43A [7]

IBT-2 adalah jembatan H arus tinggi terintegrasi penuh untuk aplikasi penggerak motor menggunakan *Infineon* BTS7960 setengah jembatan arus tinggi. BTS7960 adalah bagian dari keluarga *NovalithICTM* yang mengandung satu MOSFET high-channel pchannel dan MOSFET lowside channel dengan IC driver terintegrasi dalam satu paket. Karena saklar *high-channel*, kebutuhan untuk pompa pengisian dihilangkan sehingga meminimalkan EMI. *Interfacing* ke mikrokontroler dipermudah oleh IC driver terintegrasi yang menampilkan input level logika, diagnosis dengan indra saat ini, penyesuaian laju perubahan tegangan, pembuatan waktu mati dan perlindungan terhadap suhu berlebih, tegangan lebih, *underVoltage*, arus berlebih, dan korsleting. IBT-2 memberikan solusi yang dioptimalkan biaya untuk *drive motor* PWM arus tinggi yang dilindungi dengan konsumsi ruang papan yang sangat rendah.

Spesifikasi:

- Tegangan Pengoperasian 5,5 hingga 27V (B +)
- Jalur resistensi dari *typ.* 16 mOhm pada 25 ° C
- Arus diam rendah dari *typ.* 7 uA pada 25 ° C
- Kemampuan PWM hingga 25 kHz dikombinasikan dengan *freewheeling* aktif
- Batasan arus mode sakelar untuk mengurangi disipasi daya dalam arus lebih
- Level batasan saat ini dari 43 A *typ.*
- Diagnosis bendera status dengan kemampuan akal saat ini
- Suhu lebih rendah ditutup dengan perilaku kait
- Kunci tegangan lebih besar
- Tegangan rendah ditutup
- Sirkuit *driver* dengan input level logika

Bentuk fisik dari *Driver motor* BTS7960 43A dapat dilihat pada Gambar 2.8



Gambar 2.8 *Driver Motor* BTS7960 43A

2.8 *Spring* Tekan [8]

Pegas (*spring*) adalah perangkat fleksibel yang digunakan untuk menyimpan energi elastis dan melepaskannya ketika dibutuhkan. Gaya yang dihasilkan oleh Pegas bisa bersifat tekan (compressive) atau tarik (tensile) dan bisa juga linear atau radial. Contoh *spring* tekan dapat dilihat pada Gambar 2.9



Gambar 2.9 *Spring* Tekan

Spring tekan yang digunakan pada anjang bangun alat ini menggunakan bahan besi berukuran 3mm dengan diameter dalam *spring* 10mm dengan tinggi 100mm.

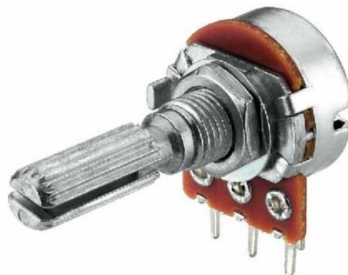
Spring tekan digunakan sebagai media dalam meneruskan getaran yang dihasilkan oleh motor menuju papan pengujian yang akan dipergunakan sebagai tempat menguji alat.

2.9 *Potensiometer* [9]

Potensiometer (POT) adalah salah satu jenis Resistor yang Nilai Resistansinya dapat diatur sesuai dengan kebutuhan Rangkaian Elektronika ataupun kebutuhan pemakainya. *Potensiometer* merupakan Keluarga Resistor yang tergolong dalam Kategori Variable Resistor. Secara struktur, *Potensiometer* terdiri dari 3 kaki Terminal dengan sebuah shaft atau tuas yang berfungsi sebagai pengaturnya.

Potensiometer yang digunakan pada alat uji menggunakan jenis *potensiometer rotary* yaitu *potensiometer* yang nilai resistansinya dapat diatur dengan cara memutarakan *Wiper* sepanjang lintasan yang melingkar. *Potensiometer Rotary* sering disebut juga dengan Thumbwheel Potentiometer.

Sebuah *Potensiometer* (POT) terdiri dari sebuah elemen *resistif* yang membentuk jalur (*track*) dengan terminal dikedua ujungnya. Sedangkan terminal lainnya ditengah adalah Penyapu (*Wiper*) yang dipergunakan untuk menentukan pergerakan pada jalur elemen *Resistive*. Pergerakan Penyapu (*Wiper*) pada Jalur Elemen *Resistif* inilah yang mengatur naik-turunnya Nilai Resistansi sebuah *Potensiometer*. Bentuk fisik dari *potensiometer* dapat dilihat pada Gambar 2.10



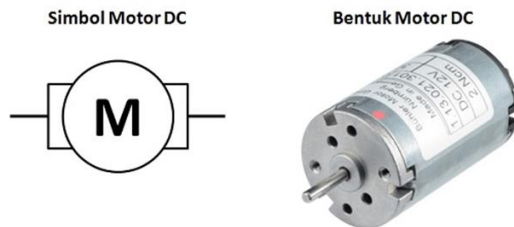
Gambar 2.10 *Potensiometer 1K Ohm*

2.10 Motor DC 24 Volt[10]

Motor Listrik DC atau *DC Motor* adalah suatu perangkat yang mengubah energi listrik menjadi energi kinetik atau gerakan (*motion*). Motor DC ini juga dapat disebut sebagai motor arus searah. Seperti

namanya, DC Motor memiliki dua terminal dan memerlukan tegangan arus searah atau DC (*Direct Current*) untuk dapat menggerakkannya. Motor Listrik DC ini biasanya digunakan pada perangkat-perangkat Elektronik dan listrik yang menggunakan sumber listrik DC seperti Vibrator Ponsel, Kipas DC dan Bor Listrik DC.

Motor Listrik DC atau *DC Motor* ini menghasilkan sejumlah putaran per menit atau biasanya dikenal dengan istilah RPM (*Revolutions per minute*) dan dapat dibuat berputar searah jarum jam maupun berlawanan arah jarum jam apabila polaritas listrik yang diberikan pada Motor DC tersebut dibalik. Motor Listrik DC tersedia dalam berbagai ukuran rpm dan bentuk. Motor DC yang digunakan pada alat ini mempunyai spesifikasi tegangan sebesar 24Vdc dengan kecepatan 2400 rpm. Apabila tegangan yang diberikan ke Motor Listrik DC lebih rendah dari tegangan operasionalnya maka akan dapat memperlambat rotasi motor DC tersebut sedangkan tegangan yang lebih tinggi dari tegangan operasional akan membuat rotasi motor DC menjadi lebih cepat. Motor yang digunakan pada alat ini adalah motor DC shunt. Motor DC tipe Shunt adalah Motor DC yang kumparan medannya dihubungkan secara paralel dengan kumparan angker (*armature winding*). Motor DC tipe Shunt ini merupakan tipe Motor DC yang sering digunakan, hal ini dikarenakan Motor DC Shunt memiliki kecepatan yang hampir konstan meskipun terjadi perubahan beban (kecepatan akan berkurang apabila mencapai torsi (*torque*) tertentu).. Bentuk fisik motor DC dapat dilihat pada Gambar 2.11



Gambar 2.11 Motor DC 24 Volt

2.11 Kontrol Kecepatan pada Motor DC [11]

Motor DC yang digunakan pada alat ini adalah motor DC dengan magnet permanen. Motor DC jenis ini memiliki dua buah magnet permanen sehingga timbul medan magnet di antara kedua magnet tersebut. Di dalam medan magnet inilah jangkar/rotor berputar. Jangkar yang terletak di tengah motor memiliki jumlah kutub yang

ganjil dan pada setiap kutubnya terdapat lilitan. Lilitan ini terhubung ke area kontak yang disebut komutator. Sikat (*brushes*) yang terhubung ke kutub positif dan negatif motor memberikan daya ke lilitan sedemikian rupa sehingga kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya, sedangkan lilitan lain akan ditarik ke magnet permanen yang lain sehingga menyebabkan jangkar berputar. Ketika jangkar berputar, komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga jangkar akan terus berputar selama kutub positif dan negatif motor diberi daya. rumus persamaan dari kecepatan putar motor DC (N).

$$N = \frac{V_{TM} - I_A R_A}{K_{\emptyset}} \dots\dots\dots (2.1)$$

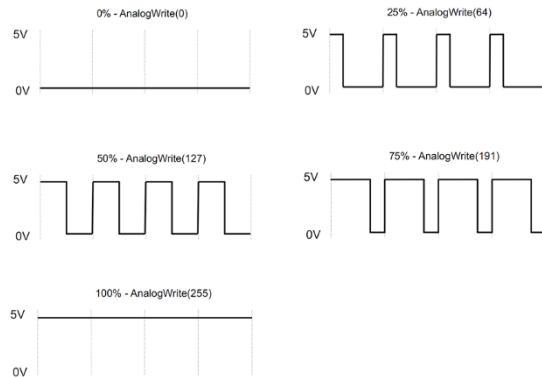
Keterangan:

- V_{TM} : Tegangan terminal
- I_A : Arus jangkar motor
- R_A : Hambatan jangkar motor
- K : Konstanta motor
- $\emptyset$: Fluk magnet yang terbentuk pada motor

Pengendalian kecepatan putar motor DC dapat dilakukan dengan mengatur besar tegangan terminal motor V_{TM} . Untuk mengatur besar tegangan yang masuk pada motor DC yaitu dengan mengatur PWM (*Pulse Width Modulation*) pada Arduino yang nantinya masuk ke driver motor BTS7960. Pada Arduino, sinyal PWM adalah sinyal yang beroperasi pada frekuensi 500Hz. Pada board Arduino Uno, pin yang bisa dimanfaatkan untuk PWM adalah pin yang diberi tanda (~), yaitu pin 3, 5, 6, 9, 10, dan pin 11.

PWM pada Arduino bekerja pada frekuensi 500Hz, artinya 500 siklus/ketukan dalam satu detik. Untuk setiap siklus, kita bisa memberi nilai dari 0 hingga 255. Ketika kita memberikan angka 0, berarti pada pin tersebut tidak akan pernah bernilai 5 Volt (pin selalu bernilai 0 Volt). Sedangkan jika kita memberikan nilai 255, maka sepanjang siklus akan bernilai 5 Volt (tidak pernah 0 Volt). Jika kita memberikan nilai 127 (setengah dari 0 hingga 255, atau 50% dari 255), maka setengah siklus akan bernilai 5 Volt, dan setengah siklus lagi akan bernilai 0 Volt. Sedangkan jika jika memberikan 25% dari 255 ($1/4 * 255$ atau 64), maka $1/4$ siklus akan bernilai 5 Volt, dan $3/4$

sisanya akan bernilai 0 Volt, dan ini akan terjadi 500 kali dalam 1 detik. Untuk visualisasi siklus PWM dapat dilihat pada Gambar 2.13



Gambar 2.12 Siklus Sinyal PWM pada Arduino

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

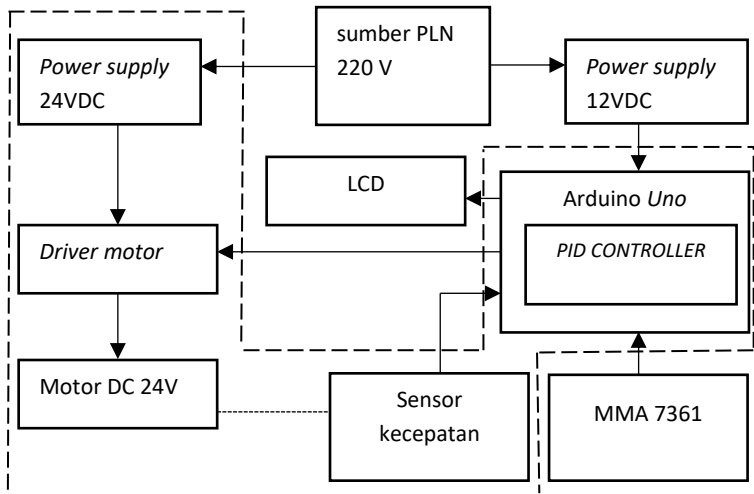
BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai perancangan dan pembuatan alat “Kontrol Kecepatan Motor DC dengan Beban Bandul”. 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* alat uji getaran beserta perangkat elektriknya. Dan perancangan perangkat lunak (*software*) meliputi pemrograman Arduino IDE pada Arduino Uno untuk pengukuran kecepatan motor DC pada sensor *Optocoupler* serta kontrol PID kecepatan motor DC.

3.1 Diagram Fungsional Sistem

Pada perancangan dan pembuatan Proyek Akhir yang berjudul “Kontrol Kecepatan Motor DC dengan Beban Bandul” ini akan dijelaskan diagram fungsional sistem dari perancangan alat yang akan dibuat pada Gambar 3.1



Gambar 3.1 Diagram Fungsional Alat Uji Getaran

Sumber awal menggunakan sumber 220VAC yang dirubah menjadi tegangan DC dalam 2 *power supply switching*, yang pertama

power supply switching dengan keluaran 24VDC 10A yang akan menjadi masukan motor 24V, yang kedua *power supply switching* dengan keluaran 12VDC yang menjadi masukan Arduino Uno.

Getaran yang timbul pada alat ini dihasilkan oleh manipulasi gerakan motor dengan mengatur kecepatan motor yang diberi beban bandul dan kemudian hasil getaran motor akan diterima oleh *spring* tekan yang akan melanjutkan getaran menuju tempat pengujian alat yang berupa plat besi. Pada bagian bawah plat besi terdapat 2 sensor yakni sensor MMA 7361 untuk mengambil data percepatan getaran yang terjadi pada plat besi, dan sensor optocoupler TCST1202 untuk mengetahui kecepatan putar motor DC. Data percepatan dan kecepatan motor yang diterima oleh sensor kemudian diolah dalam mikrokontroler Arduino Uno, hasil dari sensor MMA 7361 yang berupa percepatan akan dikonversi menjadi getaran dengan satuan hertz (Hz) dan hasil sensor *Optocoupler* TCST1202 akan diolah dengan kontrol PID sehingga mendapat hasil terbaik sesuai dengan yang diinginkan dan selanjutnya akan ditampilkan pada (LCD)

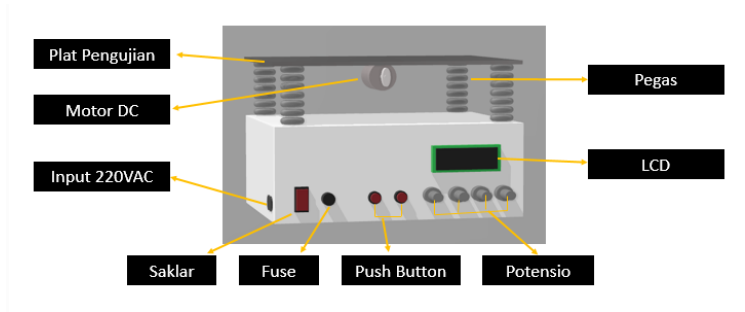
Adapun tampilan (LCD) yang disajikan diantaranya adalah nilai getaran yang dihasilkan oleh alat, kecepatan motor dan nilai Kp, Ki, Kd, serta set point yang diatur oleh *potensiometer* pada alat ini.

3.2 Perancangan Perangkat Keras (*Hardware*)

Pada perancangan *hardware* pada Proyek Akhir ini yang dibahas terdiri dari perancangan *prototype* alat uji getaran dan perancangan elektrik berupa rangkaian sensor kecepatan dan kontrol PID.

3.2.1 Perancangan *Prototype* Alat Uji Getaran

Pada pengerjaan Proyek Akhir ini, setelah berbagai macam rangkaian yang telah disebutkan di atas telah dikerjakan, langkah selanjutnya yakni membuat perancangan mekanik yang sesuai dengan pengerjaan Proyek Akhir ini. Adapun mekanik yang dibuat sedemikian rupa dengan tujuan menyesuaikan dengan kondisi yang dibutuhkan. Adapun desain mekanik yang telah dibuat yakni modul alat uji getaran yang telah menyatu dengan tempat pengujian getaran. Gambar perancangan Alat Uji Getaran seperti pada Gambar 3.2



Gambar 3.2 Perancangan Alat Uji Getaran

Dari Gambar 3.2 ditunjukkan rancangan Alat Uji Getaran beserta ukuran dari mekanik yang akan dibuat. Ukuran tersebut di desain sedemikian rupa agar menghasilkan mekanik Alat Uji Getaran. Komponen-komponen yang digunakan pada Alat Uji Getaran diletakkan sesuai dengan fungsinya agar Alat Uji Getaran dapat digunakan dengan mudah. Setelah melakukan proses perancangan Alat Uji Getaran tersebut, selanjutnya merupakan proses pembuatan. Pembuatan mekanik Alat Uji Getaran mengacu pada perancangan mekanik Alat Uji Getaran yang telah dibuat. Foto hasil dari pembuatan mekanik Alat Uji Getaran dapat dilihat pada Gambar 3.2

Desain mekanik terbuat dari bahan akrilik dengan ketebalan 5mm, sehingga modul alat uji getaran kuat menahan beban getaran yang dihasilkan dan beban yang akan diuji. Bentuk fisiknya juga terdapat bagian yang ditujukan untuk pegas tekan dan motor DC yang diletakkan ditengah plat pengujian untuk menghasilkan getaran. Berikut adalah bentuk fisik dari mekanik yang telah dibuat. Perancangan mekanik yang telah dibuat ini secara keseluruhan pada akhirnya didapatkan desain yang baik dan tidak terkesan hanya sebatas sebagai alat untuk pengerjaan Proyek Akhir saja tetapi juga dapat langsung diterapkan pada kehidupan sehari-hari. Walaupun masih perlu adanya penyempurnaan dikemudian hari.

Perancangan *hardware* pada Proyek Akhir ini terdiri dari perancangan *prototype* alat uji getaran dengan menggunakan sensor MMA 7361 sebagai sensor percepatan dan sensor *Optocoupler* untuk pengaturan kecepatan motor untuk menghasilkan getaran. Pada alat ini juga terdapat *potensiometer* untuk mengatur nilai K_p , K_i , K_d dan *setpoint* sebagai masukan nilai kontrol PID.

Perancangan prototype alat uji getaran mempunyai batasan beban sebesar 5Kg. Perancangan desain alat uji getaran dibuat menggunakan *software* sketchup.

Spesifikasi alat uji getaran :

Modul

Panjang : 40 cm

Lebar : 30 cm

Tinggi : 15 cm

Volume Modul : 18,000 cm³

Plat Besi

Panjang : 40 cm

Lebar : 30 cm

Luas : 120 cm²

Pegas

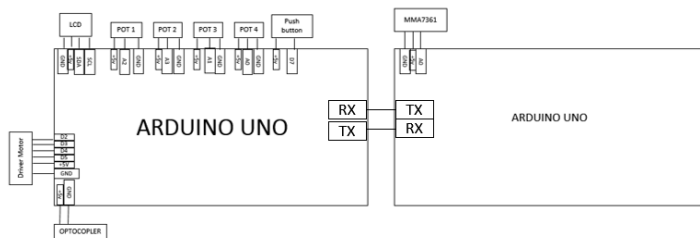
Diameter : 30 mm

Tebal : 3 mm

Tinggi : 100 mm

3.2.2 Rangkaian Alat Uji Getaran

Pada Gambar 3.3 adalah rangkaian alat uji getaran secara keseluruhan.



Gambar 3.3 Rangkaian Alat Uji Getaran

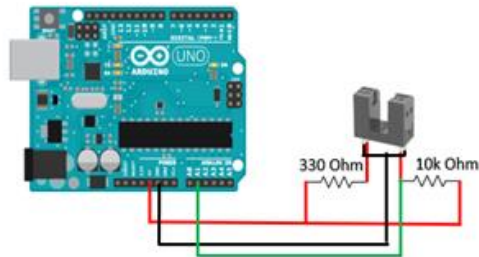
Pada Gambar 3.3 rangkaian alat uji getaran menggunakan 2 Arduino dengan Arduino pertama dihubungkan dengan *Driver motor* BTS7960, LCD 16x2 I2C, 4 *potensiometer*, *push button*, dan sensor *Optocoupler* TCST1202. Sedangkan Arduino lainnya dihubungkan pada sensor getaran MMA7361.

Untuk wiring *Driver motor* BTS7960 dihubungkan pada pin digital Arduino pin 2,3,4 dan 5. Pada LCD 16x2 I2C dihubungkan pada pin SCL SDA pada Arduino, sedangkan 4 *potensiometer* dihubungkan pada pin analog Arduino A2,A3,A0 dan A1. Semua

komponen membutuhkan VCC dan GND dari Arduino. Untuk sensor getaran MMA7361 dihubungkan pada Arduino kedua pada pin A0, VCC dan GND. Untuk pengiriman data sensor getaran ke Arduino pertama menggunakan komunikasi serial TX RX sehingga dapat ditampilkan pada LCD 16x2 secara bersamaan.

3.2.3 Rangkaian *Optocoupler* Sensor

Pada Gambar 3.4 adalah rangkaian sensor *Optocoupler* pada alat penguji getaran.

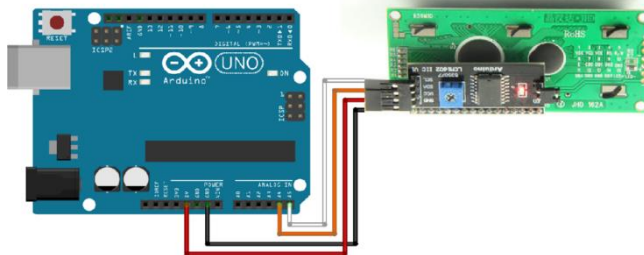


Gambar 3.4 Rangkaian *Optocoupler* Sensor Ke Arduino Uno

Pada Gambar 3.4 terdapat gambar rangkaian sensor optokopler ke Arduino Uno. Dapat dilihat bahwa terdapat empat kaki pada sensor. Garis merah pada gambar memperlihatkan bahwa kaki sensor terhubung dengan vcc (5Volt), garis hijau terhubung dengan pin analog A1, sedangkan garis hitam terhubung dengan pin GND.

3.2.4 Rangkaian Tampilan Pada LCD (*Liquid Crystal Display*)

Pada Gambar 3.5 adalah rangkaian LCD 16x2 I2C ke Arduino pada alat penguji getaran

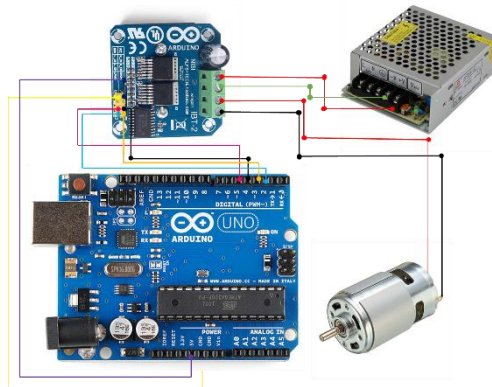


Gambar 3.5 Rangkaian LCD 16x2 I2C ke Arduino Uno

Pada Gambar 3.5 terdapat gambar rangkaian LCD 16x2 I2C ke Arduino Uno. Dapat dilihat bahwa terdapat empat kaki pada LCD 16x2 I2C. Garis merah pada gambar memperlihatkan bahwa kaki sensor terhubung dengan VCC (5Volt), sedangkan garis hitam terhubung dengan pin GND dan 2 kaki lainnya terhubung pada pin SDA SCL pada Arduino Uno.

3.2.5 Rangkaian *Driver Motor* BTS7960 43A

Pada Gambar 3.6 adalah rangkaian *Driver motor* BTS7960 43A pada alat penguji getaran.



Gambar 3.6 Rangkaian *Driver Motor* BTS7960 43A

Pada Gambar 3.6 terdapat gambar rangkaian *Driver motor* BTS7960 43A yang dihubungkan dengan Arduino, *power supply switching* dan motor dc 24 Volt. Dapat dilihat bahwa terdapat enam kaki *Driver motor* yang dihubungkan ke Arduino Uno yaitu kaki LPWM yang dihubungkan ke pin digital 2 Arduino, RPWM yang dihubungkan ke pin digital 3 Arduino, kaki REN yang dihubungkan ke pin digital 4 Arduino dan yang terakhir kaki LEN yang dihubungkan ke pin digital 5 Arduino. Sedangkan pada terminal yang terdapat pada *Driver motor* BTS7960 43A dihubungkan pada *power supply switching* 24 Volt positif ditunjukkan dengan garis warna merah dan negative ditunjukkan garis berwarna hitam. Terminal yang lain dihubungkan dengan positif yang ditunjukkan dengan garis warna merah dan negatif yang ditunjukkan garis berwarna hitam ke motor DC 24 Volt. *Power supply switching* dihubungkan langsung ke sumber tegangan 220 VAC.

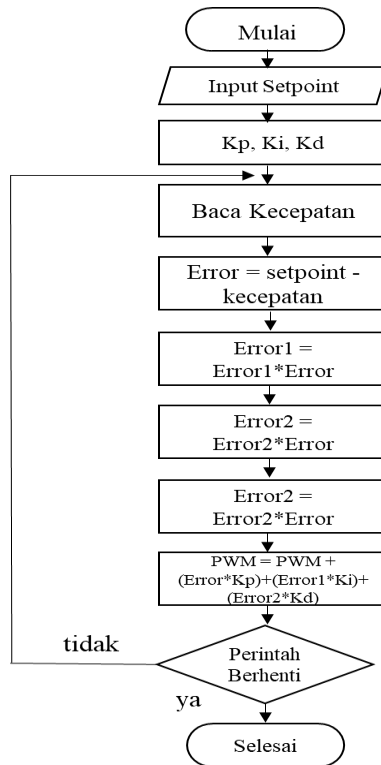
3.3 Perancangan Perangkat Lunak (Software)

Pada perancangan perangkat *software* pada Proyek Akhir ini yang dibahas terdiri dari pemrograman pengukuran kecepatan motor dengan menggunakan sensor *Optocoupler* pada *software* Arduino IDE.

3.3.1 Pemrograman Pembacaan Sensor *Optocoupler*

Pada alur kerja dapat dilihat alur kerja dari pembacaan sensor *Optocoupler*. Pada pembacaan ini menjelaskan langkah langkah pembacaan sensor *Optocoupler* dari memasukan nilai K_p , K_i dan K_d hingga menampilkan pembacaan sensor *Optocoupler* ke LCD 16x2.

Pada pemrograman pembacaan sensor *Optocoupler* TCST1202 ini menggunakan *PID library basics* dari Arduino IDE seperti pada Gambar 3.7



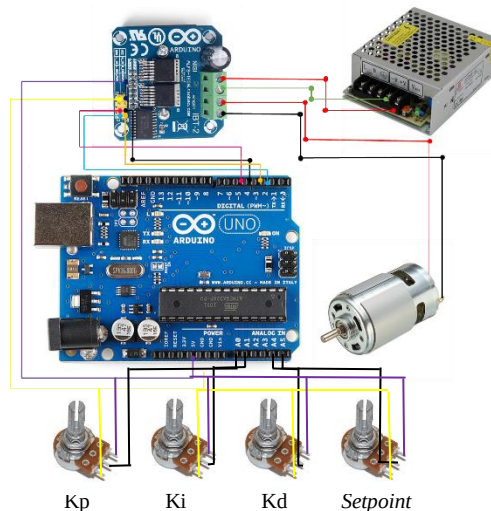
Gambar 3.7 Flowchart Pembacaan Sensor *Optocoupler*

Pada Gambar 3.7 terdapat gambar urutan pembacaan sensor *Optocoupler* TCST2102 yang memiliki urutan dari memasukan nilai K_p, K_i, K_d dan Setpoint sebagai kontroler utama system ini yaitu kontrol PID. Lalu saat motor mulai berputar sensor *Optocoupler* TCST2102 akan membaca kecepatan motor yang dihasilkan. Kemudian hasil pembacaan sensor *Optocoupler* TCST2102 diolah oleh Arduino Uno sesuai dengan nilai K_p, K_i , dan K_d yang sudah dimasukan dari *potensiometer* pada alat ini.

Pembuatan program kontroler PID ini dilakukan berdasarkan persamaan kontroler PID digital. Kontroler PID digital merupakan bentuk lain dari kontroler PID yang diprogram dan dijalankan menggunakan komputer atau mikrokontroler. *Error* dan *last_error* yang akan digunakan pada perhitungan aksi kontroler PID. Setiap satu kali *looping* program, *error* akan diperbaharui dengan data yang diambil dari sensor, dan sebelumnya akan disimpan di *last_error*. Keluaran dari perhitungan program kontroler PID ini adalah nilai PWM

3.4 Penjelasan Kontrol Kecepatan Motor

Pada Gambar 3.8 adalah rangkaian kontrol kecepatan motor dc 24V yang terdiri dari Arduino Uno, *Driver Motor* BTS7960, *Power Supply* 24V, Motor DC 24V dan *Potensiometer*. Rangkaian kontrol kecepatan pada alat uji getar seperti pada Gambar 3.8



Gambar 3.8 Rangkaian Kontrol Kecepatan Motor DC

Pada Gambar 3.8 menjelaskan bahwa rangkaian kontrol kecepatan yang terdiri dari Arduino Uno, *Driver Motor* BTS7960, *Power Supply* 24V, Motor DC 24V dan *Potensiometer*. Pada alat ini kontrol kecepatan motor DC menggunakan metode modulasi lebar pulsa atau dalam bahasa inggris disebut *pulse width modulation* (PWM). PWM dihasilkan dari mikrokontroler Arduino Uno yang memiliki alokasi data 8 bit atau memiliki rentang data dari 0-255. Motor DC yang digunakan pada alat ini sebesar 24V dengan kecepatan 2400 rpm. Arduino digunakan sebagai kontroler yang digunakan untuk mengatur dan menghasilkan pwm yang masuk pada *driver motor* BTS7960, *driver motor* diperlukan untuk memodulasi tegangan yang masuk Arduino untuk disesuaikan dengan tegangan pada motor 24 Volt.

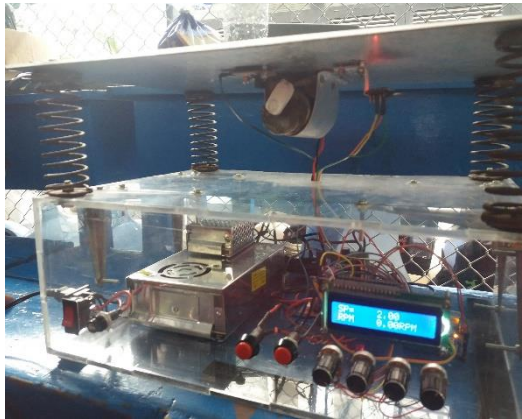
Untuk mengatur kecepatan motor DC pada alat ini menggunakan kontrol PID. Pada alat ini juga terdapat 4 *potensiometer* yang digunakan untuk mengatur nilai Kp, Ki, Kd dan setpoint pada kontrol PID. Nilai yang didapat dari *potensiometer* nantinya akan diolah pada Arduino dan hasilnya akan diteruskan ke driver motor berupa tegangan untuk mengatur kecepatan motor DC.

-----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 analisa dapat diketahui kelemahan-kelemahan dari sistem, sehingga dapat dilakukan perbaikan, pengembangan, dan penyempurnaan sistem. Alat yang telah terealisasi dapat dilihat pada Gambar 4.1



Gambar 4.1 Foto Keseluruhan Alat

Dalam bab ini dibahas pengujian serta analisa dari perancangan sistem yang telah dibuat pada Proyek 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 LCD 16x2 I2C

Untuk menguji LCD ini digunakan program sederhana untuk menampilkan karakter pada tiap baris LCD. Pertama dilakukan pengujian apakah LCD dapat menampilkan karakter sesuai dengan lokasi peletakan karakter yang dimaksud menggunakan *library* I2C pada *software* Arduino IDE. Program yang digunakan seperti Gambar 4.2

```

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);
void loop()
if (count == 1000)
{
    lcd.clear();
    lcd.setCursor (0, 0);
    lcd.print("SP=");
    lcd.setCursor (7, 0);
    lcd.print(Setpoint);
    lcd.setCursor (0, 1);
    lcd.print("RPM");
    lcd.setCursor (7, 1);
    lcd.print(rpm);
}

```

Gambar 4.2 Program Tampilan LCD 16x2 I2C

Dari program pada Gambar 4.2 LCD 16x2 I2C menampilkan tampilan karakter seperti pada Gambar 4.3.



Gambar 4.3 Tampilan LCD 16x2 I2C

4.2 Pengujian Sensor *Optocoupler* TCST 1202

Pada pengujian sensor TCST 1202 ini dilakukan untuk mengetahui keakuratan dari sensor *Optocoupler* TCST 1202 ini. Penempatan sensor *Optocoupler* TCST 1202 juga berpengaruh pada hasil pembacaan sensor. Penempatan sensor harus tepat membaca pul putaran dari motor. Foto penempatan sensor *Optocoupler* seperti pada Gambar 4.4.



Gambar 4.4 Penempatan Sensor *Optocoupler* TCST1202

Dari penempatan sensor *Optocoupler* TCST1202 dapat diperoleh data yang diinginkan sehingga dapat dibandingkan dengan alat ukur kecepatan motor yaitu tachometer. Dari data ini dapat diketahui *error* dari pembacaan sensor *Optocoupler* TCST 1202 sehingga dapat mengetahui tingkat keakuratan sensor.

Tabel 4.1 Kalibrasi Sensor *Optocoupler*

No	Kecepatan Pada Sensor <i>Optocoupler</i> (Rpm)	Kecepatan Pada Tachometer (Rpm)	<i>Error</i> (%)
1	224	245	9,375
2	476	490	2,941176
3	749	786	4,93992
4	1089	1139	4,591368
5	1341	1387	3,430276
6	1612	1658	2,853598
7	1830	1899	3,770492
8	1933	1978	2,327988
9	2213	2308	4,292815
10	2322	2368	1,981051

Pada Tabel 4.1 menunjukan data kalibrasi sensor *Optocoupler* TCST 1202 dengan *tachometer*. Hasil pengukuran menunjukan bahwa *error* maksimal yaitu 9,375% dengan keadaan tanpa beban.

4.3 Pengujian Motor DC dengan Open Loop Control

Pengujian ini dilakukan pada rangkaian open loop kontrol kecepatan motor DC dengan menggunakan sensor *Optocoupler* TCST 1202 dan *potensiometer* yang diputar secara manual. Pengujian ini dilakukan untuk mengetahui pengaruh beban terhadap kecepatan motor dan juga untuk menentukan beban maksimal yang dapat diuji pada alat ini, serta kecepatan maksimal motor DC yang dapat dihasilkan alat ini. Hasil pengujian open loop test dapat dilihat pada Tabel 4.2

Tabel 4.2 Data Kecepatan Motor DC dengan *Open Loop Control*

No	Tegangan Input Potensio (Volt)	Kecepatan Pada Sensor <i>Optocoupler</i> (RPM)					
		Tanpa Beban	1Kg	2Kg	3Kg	4Kg	5Kg
1	0,46	224	224	222	221	218	217
2	0,85	476	474	471	465	457	453
3	1,29	749	749	749	749	746	745
4	1,85	1089	1083	1052	1043	1040	1032
5	2,28	1341	1329	1324	1322	1329	1323
6	2,74	1612	1612	1602	1603	1572	1592
7	3,14	1830	1817	1822	1817	1700	1864
8	3,34	1933	1925	1901	1912	1928	1914
9	3,39	2213	2233	2261	2203	2201	2214
10	5	2322	2313	2310	2307	2309	2311

Dari data pengujian pada Tabel 4.2 dapat diketahui bahwa berat beban mempengaruhi kecepatan motor yang dibaca oleh sensor walaupun pengaruh beban tidak terlalu besar. Sedangkan kecepatan maksimal motor DC yang dapat dihasilkan alat ini yaitu 2322 RPM pada saat keadaan tanpa beban dengan tegangan tegangan input pada

potensiometer 5 Volt. Untuk berat maksimal beban yang dapat di uji pada alat ini yaitu 5 Kg. Karena pada saat pengujian beban lebih dari 5 Kg per pegas pada alat ini terlalu rapat dan miring sehingga getaran yang dihasilkan tidak maksimal.

4.4 Pengujian Karakteristik Motor DC

Pengujian dilakukan untuk mengetahui karekteristik motor DC, pengujian ini mengukur nilai tegangan yang masuk pada motor DC, dan kecepatan yang dihasilkan oleh motor DC. Hasil dari pengujian ini digunakan untuk mengetahui kecepatan motor terhadap nilai pda potensio yang diberikan. Hasil pengujian karakteristik motor dc seperti pada Gambar 4.3

Tabel 4.3 Pengujian Karakteristik Motor DC

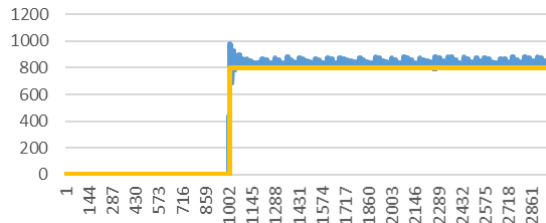
No	Vin (Volt)	Kecepatan (Rpm)	Arus (Ampere)
1	2,2	226	0,23
2	4,3	466	0,53
3	6,1	755	0,66
4	6,7	1092	0,77
5	11,3	1260	1,16
6	13,1	1635	1,29
7	15,2	1835	1,46
8	16,1	1942	1,59
9	16,5	2221	1,61
10	24	2322	1,83

Berdasarkan pada Tabel 4.3 didapatkan data nilai Vin, Rpm, dan Arus. Data pada Tabel 4.3 digunakan untuk mengetahui kecepatan motor pada tegangan 2,2 Volt hingga 24 Volt. Dari Tabel 4.3 dapat diketahui bahwa kecepatan maksimal adalah 2322rpm dengan tegangan masukan sebesar 24 Volt. Sedangkan ketika dilakukan pengukuran dengan nilai tegangan 2,2 Volt nilai kecepatan motor terukur sebesar 226rpm.

4.5 Pengujian Motor DC dengan *Close Loop Control*

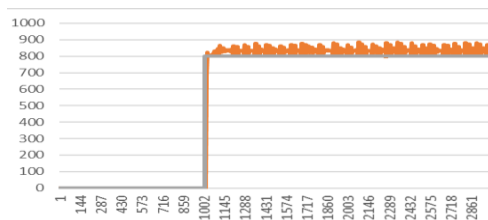
Pengujian ini dilakukan pada rangkaian kontrol PID kecepatan motor DC dengan menggunakan sensor *Optocoupler* TCST 1202 dengan set point 800 rpm. Untuk merancang sistem kontrol PID, kebanyakan dilakukan dengan metoda coba-coba atau (*trial&error*). Hal ini disebabkan karena parameter Kp, Ki, dan Kd tidak *independent*, untuk mendapatkan hasil kontrol yang baik, diperlukan

langkah coba-coba dengan kombinasi antara P, I, dan D sampai ditemukan nilai K_p , K_i , dan K_d seperti yang diinginkan. Hasil kontrol PID dengan $K_p:1$, $K_i:0$, $K_d:0$ ditunjukkan pada Gambar 4.5



Gambar 4.5 Kontrol PID dengan $K_p:1$, $K_i:0$, $K_d:0$

Pada Gambar 4.5 dapat diketahui bahwa nilai K_p , K_i , dan K_d yang digunakan dalam pengambilan data adalah $K_p:1$, $K_i:0$, dan $K_d:0$. Dapat dilihat dengan ketiga nilai yang dimasukkan saat pengambilan data grafik seperti Gambar 4.5 dengan nilai *overshoot* sebesar 985rpm. Hasil kontrol PID dengan $K_p:2$, $K_i:0$, $K_d:0$ ditunjukkan pada Gambar 4.6



Gambar 4.6 Kontrol PID dengan $K_p:2$, $K_i:0$, $K_d:0$

Pada Gambar 4.6 dapat diketahui bahwa nilai K_p , K_i , dan K_d yang digunakan adalah sebesar $K_p:2$, $K_i:0$, $K_d:0$. Pada percobaan kali ini terdapat perbedaan dengan percobaan yang menghasilkan grafik pada Gambar 4.5, diketahui pada Gambar 4.5 nilai K_p yang digunakan bernilai 1 sedangkan nilai K_p yang digunakan pada percobaan yang menghasilkan grafik pada Gambar 4.6 adalah sebesar 2. Pada Gambar 4.6 terdapat *overshoot* dengan nilai 800rpm. Jika dibandingkan dengan grafik pada Gambar 4.5 terdapat perbedaan nilai sebesar 185 rpm lebih kecil pada grafik Gambar 4.6

4.6 Pengujian Kecepatan Motor DC Terhadap Getaran

Pengujian ini dilakukan untuk mengetahui pengaruh kecepatan motor terhadap getaran yang dihasilkan alat ini dengan satuan Hertz. Sensor getaran yang digunakan pada alat ini yaitu MMA 7361. Pada pengujian ini data diambil dengan cara acak yang berjumlah 10 data dari kecepatan rendah sampai kecepatan maksimal motor DC.

Tabel 4.4 Kecepatan Motor Terhadap Getaran

No	Kecepatan Pada Sensor <i>Optocoupler</i> (Rpm)	Getaran Pada Sensor MMA 7361 (Hz)
1	224	93
2	476	201
3	749	322
4	1089	453
5	1341	570
6	1612	687
7	1830	771
8	1933	832
9	2213	946
10	2322	987

Pada Tabel 4.4 menjelaskan bahwa kecepatan motor dan getaran yang dihasilkan berbanding lurus. Sehingga semakin besar nilai kecepatan motor semakin besar pula getaran yang dihasilkan oleh alat ini. Getaran maksimal yang bisa dihasilkan alat ini yakni 987Hz pada kecepatan motor maksimal yaitu 2322 rpm.

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

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 Proyek Akhir ini.

5.1 Kesimpulan

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

1. Dari data yang diambil dapat diketahui nilai maksimal kecepatan putar dari motor DC yakni 2322 rpm
2. Nilai frekuensi maksimal yang dihasilkan alat ini yakni 987Hz
3. Objek yang diuji maksimal 5 kg
4. Hasil kontrol PID yang didapatkan yaitu pada nilai $K_p=2$, $K_i=0$, $K_d=0$ pada *set point* 800 rpm dengan beban bandul.

5.2 Saran

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

1. Penambahan sensor yang lebih baik
2. Perhitungan penempatan motor serta desain mekanik
3. Penambahan HMI yang sesuai dengan alat ini

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

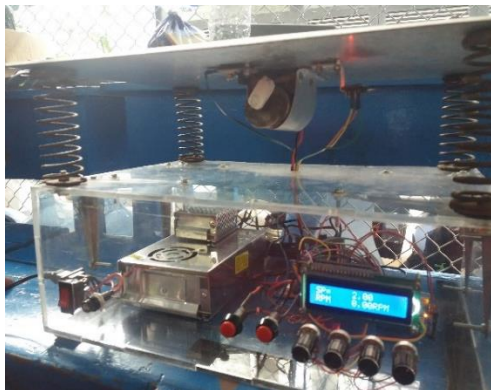
DAFTAR PUSTAKA

- [1] Andi. **Dasar–Dasar Getaran Mekanis**. Yogyakarta. Tungga Bhimadi Karya. 2001
- [2] Putra, A, E. **Belajar Mikrokontroler Teori dan Aplikasi**. Yogyakarta. Gava Media. 2002
- [3] V. N. Febrianto. Aplikasi Kontrol PID untuk Pengaturan Putaran Motor DC pada Alat Pengerpes Adonan Roti (*Screw Conveyor*). **Tugas Akhir**. Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya. 2014
- [4] Bagaskara. **Pengertian LCD dan Fungsinya**. 2013 (<http://baskarapunya.blogspot.co.id/2013/01/liquid-crystal-display-lcd-16-x-2.html>, diakses tanggal 12 Januari 2019)
- [5] Effendi, Ilham. **Pengertian dan Kelebihan Arduino**. 2014. (<http://www.it-jurnal.com/2014/05/pengertian-dan-kelebihan-Arduino.html>, diakses tanggal 12 Januari 2019)
- [6] Dudi, Rudiawan. **Pengertian dan Fungsi Power Supply**. 2014. (<http://dudirudiawan8.wordpress.com/2014/10/14/241/>, diakses tanggal 13 Januari 2019)
- [7] M. P. Darfyma. Kendali Kecepatan dan Posisi *Automated Guide Vehicle* Menggunakan Fuzzy Logic dan PID Control. Bandung **Tugas Akhir**. Universitas Telkom. 2016.
- [8] Albana, Muhammad Hasan. **Pengetahuan Dasar Pegas**. 2016. (<http://mhasanalbana.blogspot.com/2016/12/pengetahuan-dasar-pegas-spring.html>, diakses tanggal 13 Januari 2019)
- [9] Dickson, Kho. **Pengertian dan Fungsi Potensiometer**. 2013 (<http://teknikelektronika.com/pengertian-fungsi-potensiometer/>, diakses tanggal 12 Januari 2019)
- [10] Zuhul. **Prinsip Dasar Elektroteknik**. Jakarta. Gramedia Pustaka Utama. 2004.
- [11] Fahmizal. **Rumus Kecepatan Putar Motor DC**. 2010 (<http://fahmizaleeits.wordpress.com/tag/rumus-kecepatan-putar-motor-dc/>, diakses tanggal 12 Januari 2019)

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

LAMPIRAN A

Alat Uji Getaran



Program Kontrol PID

```
#include <LiquidCrystal_I2C.h>
#include <PID_v1.h>
#include<SoftwareSerial.h>
SoftwareSerial serial(0, 1);
String data;
int count=0;
#define PIN_OUTPUT 3
LiquidCrystal_I2C lcd(0x3F, 16, 2);
//Define Variables we'll be connecting to
double Setpoint, Input, Output;
int pot1, pot2, pot3, pot4, setpoint1;
//Specify the links and initial tuning parameters
double Kp = 1, Ki = 0 , Kd = 0 ;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
int state1 = HIGH;
int state2;
float rps;
float rpm, s;
long prevMillis = 0;
long interval = 200;
long currentTime;
long prevTime = 1;
long diffTime;
int rpm2;
int sensorvalue;
int sensorvalue2;
unsigned long startproses;
int incomingByte;

void setup()

{ Serial.begin(9600);
  serial.begin(9600);
  pinMode(2, OUTPUT); //LPWM
  pinMode(3, OUTPUT); //RPWM
  pinMode(4, OUTPUT); //REN
  pinMode(5, OUTPUT); //LEN
  digitalWrite(4, HIGH); //REN ENABLE
  digitalWrite(5, HIGH); //LEN ENABLE
```

```

pinMode(6, INPUT);
lcd.begin ();
// lcd.setCursor (0, 0);
// lcd.print ("RPM:");
// delay (5000);
// lcd.setCursor (0, 1);
// lcd.print ("SP:");
// delay (5000);
lcd.clear ();

//initialize the variables we're linked to
// Input = map(rpm2, 0, 2000, 0, 255);

//Setpoint=200;
analogWrite(2, 0);
//turn the PID on
myPID.SetMode(AUTOMATIC);
}

void loop()
{ int data2 ;

  ///Terima
  if(serial.available() > 0)
  {

    char c = serial.read();
    int dataterima = c - '0' ;
  }
  ///////////////////////////////////
  /*if (data.length() > 0) {
    data2 = data.toInt();
    Serial.println(data2);
    data = "";
  }*/

  if(count==1000)
  {
    lcd.clear();
    lcd.setCursor (0, 0);

```

```

    lcd.print("SP=");
    lcd.setCursor (7, 0);
    lcd.print(Setpoint);
    lcd.setCursor (0, 1);
    lcd.print("RPM");
    lcd.setCursor (7, 1);
    lcd.print(rpm);

}
if(count==2000)
{
    lcd.clear();
    lcd.setCursor (0, 0);
    lcd.print("Kp=");
    lcd.setCursor (4, 0);
    lcd.print(Kp);
    lcd.setCursor (9, 0);
    lcd.print("Kd=");
    lcd.setCursor (12, 0);
    lcd.print(Kd);
    lcd.setCursor (0, 1);
    lcd.print("Ki");
    lcd.setCursor (4, 1);
    lcd.print(Ki);
    lcd.setCursor (8, 1);
    lcd.print("F");
    lcd.setCursor (10, 1);
    lcd.print(dataterima);

}
if(count>2000)
{
    count=0;
}
// data = "";
// for (int i = 0; i <= 1000; i++)
// {
    sensorvalue = digitalRead(6);
    if (sensorvalue != 1)
        state1 = HIGH;
    else

```



```

    state1 = LOW;
    digitalWrite(13, state1); // as iR light is invisible for us, the led on
pin 13
    // indicate the state of the circuit.

    if (state2 != state1)
    {
        if (state2 > state1) {
            currentTime = micros();
            diffTime = currentTime - prevTime - startproses;
            rps = 1000000 / diffTime;
            rpm = 60000000 / diffTime;
            unsigned long currentMillis = millis();
            if (currentMillis - prevMillis > interval)
            {
                prevMillis = currentMillis;

            }
            prevTime = currentTime;
            unsigned long startproses = millis();
        }
        state2 = state1;
    }
    // if (i == 1000)
    // {
    //Serial.print(rpm); Serial.println("rpm");

    // }

    // }
    pot1 = analogRead(A2);
    pot2 = analogRead(A3);
    pot3 = analogRead(A0);
    pot4 = analogRead(A1);
    Kp = map(pot2, 0, 1023, 0, 100);
    Kp = pot2 * 0.1;
    Ki = map(pot3, 0, 1023, 0, 100);
    Ki = pot3 * 0.1;
    Kd = map(pot4, 0, 1023, 0, 100);
    Kd = pot4 * 0.1;

```

```

    Setpoint = map(pot1, 0, 1023, 0, 2400); // Input =
    analogRead(PIN_INPUT);
    if (pot1 == 0)
    {
        rpm = 0;
    }
    Input = rpm;
    if (Input > Setpoint)
    {
        Input = 0;
    }
    // Serial.print(Kp); Serial.print(" ");
    // Serial.print(Ki); Serial.print(" ");
    // Serial.print(Kd); Serial.print(" ");
    // Serial.print(rpm); Serial.print(" ");
    // Serial.print(Setpoint); Serial.print(" "); Serial.print(Input);
    // Serial.print(" "); Serial.println(Output);
    myPID.Compute();
    analogWrite(PIN_OUTPUT, Output);
    // lcd.setCursor(5, 1);
    // lcd.print (rpm);
    // lcd.setCursor(10, 1);
    // lcd.print (Setpoint);
    count++;
}

```

Program FFT Getaran

```

#include <Wire.h>
// #include <LCD.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <arduinoFFT.h>

// LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7); // 0x3F is the I2C
// bus address for an unmodified backpack
arduinoFFT FFT = arduinoFFT(); // bikin FFT object

// Voltage Reference: AVCC pin
#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) |
(0<<ADLAR))

```

```

const uint16_t samples = 64; //harus selalu 2 pangkat N
const double samplingFrequency = 10000.0 / 2.0; // frekuensi
sampling ADC 100us atau 10KHz -->  $f = 1/T$ 
volatile byte buffer_full = 0; // flag buffer full untuk menandakan
samel data telah terisi penuh
volatile byte index = 0; // untuk index sampel data

```

```

double vReal[samples]; // input dan output vectors
double vImag[samples]; // Input vectors untuk bagian imajiner

```

```

volatile unsigned long delta; // untuk mengetahui timing waktu antar
konversi ADC
volatile unsigned long sampleTime = 0;
char str[4];
unsigned long previousMillis = 0;

```

```

void setup() {
  SREG &= ~0x80;          // Global interrupt enable flag
  |
  Serial.begin(9600);

  // Timer/Counter 1 initialization
  // Clock source: System Clock
  // Clock value: 16000,000 kHz
  // Mode: CTC top=OCR1A
  // OC1A output: Disconnected
  // OC1B output: Disconnected
  // Noise Canceled: Off
  // Input Capture on Falling Edge
  // Timer Period: 0,1 ms atau 100us
  // Timer1 Overflow Interrupt: Off
  // Input Capture Interrupt: Off
  // Compare A Match Interrupt: Off
  // Compare B Match Interrupt: On
  TCCR1A = (0 << COM1A1) | (0 << COM1A0) | (0 << COM1B1) |
(0 << COM1B0) | (0 << WGM11) | (0 << WGM10);
  TCCR1B = (0 << ICNC1) | (0 << ICES1) | (0 << WGM13) | (1 <<
WGM12) | (0 << CS12) | (0 << CS11) | (1 << CS10);
  TCNT1H = 0x00;
  TCNT1L = 0x00;
  ICR1H = 0x00;

```

```

ICR1L = 0x00;
OCR1AH = 0x06;
OCR1AL = 0x3F;
OCR1BH = 0x00;
OCR1BL = 0x00;

// Timer/Counter 1 Interrupt(s) initialization
TIMSK1 = (0 << ICIE1) | (1 << OCIE1B) | (0 << OCIE1A) | (0 <<
TOIE1);

// ADC initialization
// ADC Clock frequency: 125,000 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: Timer1 Compare Match B
// Digital input buffers on ADC0: On, ADC1: On, ADC2: On, ADC3:
On
// ADC4: Off, ADC5: Off
DIDR0 = (1 << ADC5D) | (1 << ADC4D) | (0 << ADC3D) | (0 <<
ADC2D) | (0 << ADC1D) | (0 << ADC0D);
ADMUX = ADC_VREF_TYPE;
ADCSRA = (1 << ADEN) | (0 << ADSC) | (1 << ADIF) | (0 <<
ADIF) | (1 << ADIE) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
ADCSRB = (1 << ADTS2) | (0 << ADTS1) | (1 << ADTS0);

SREG |= 0x80; // Global interrupt enable flag

// activate LCD module
// lcd.begin(16, 2); // for 16 x 2 LCD module
// lcd.setBacklightPin(3, POSITIVE);
// lcd.setBacklight(HIGH);
//
// lcd.home(); // set cursor to 0,0
// lcd.print("FFT");

}

void loop() {
    while (buffer_full != 1); // tunggu hingga buffer penuh
    FFT.Windowing(vReal, samples, FFT_WIN_TYP_HAMMING,
FFT_FORWARD); // set mode windowing

```

```

    FFT.Compute(vReal, vImag, samples, FFT_FORWARD); //
    kalkulasi FFT
    FFT.ComplexToMagnitude(vReal, vImag, samples); // Hitung
    magnitude
    double fundamental_freq = FFT.MajorPeak(vReal, samples,
    samplingFrequency); // ambil fundamental frekuensi
    buffer_full = 0; // set buffer_full flag kembali ke 0
    index = 0; // reset index buffer

    // Serial.println(fundamental_freq); // Hz
    itoa(fundamental_freq, str, 10);

    /////kirim
    Serial.write(fundamental_freq);
    //////////////////////////////////

    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= 350) {
        previousMillis = currentMillis;

        // lcd.setCursor(0, 0);
        // lcd.print("Freq=");
        // lcd.print(fundamental_freq);
        // lcd.print(" ");

    }

    // Serial.println(delta);
}

ISR(ADC_vect) // Interrupt service routine untuk ADC
{
    // unsigned long current_time = micros(); // waktu saat ini
    // delta = current_time - sampleTime; // jeda waktu antar konversi
    ADC
    // sampleTime = current_time; // set waktu sekarang jadi waktu
    sampleTime

    if (index < samples) // jika buffer belum penuh maka tampung data
    {

```

```

    double vol_data = (double)ADCW * 5.0 / 1024; //convert analog_x-
->voltage value(v)
    vReal[index] = vol_data; // tampung data adc ke dalam buffer data
    untuk nantinya di olah oleh FFT
    vImag[index] = 0.0; //data imajiner harus di set 0.0 untuk
    menghindari kesalahan kalkulasi atau overflow
    index++; // increment index buffer
}
else {
    buffer_full = 1; // jika buffer sudah penuh maka variabel buffer_full
    di set 1 yang menandakan saatnya memproses perhitungan FFT
}

TIFR1 |= _BV(OCF1B); // clear Compare Match B interrupt agar
konversi ADC berikutnya dapat berjalan
}

ISR(TIMER1_COMPB_vect) // Timer 1 Interrupt service Routine
(Kagak jadi dipakai)
{
    // ADCSRA |= 0x40;
}

```

LAMPIRAN B

Datasheet Arduino UNO



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 Arduino, 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 Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

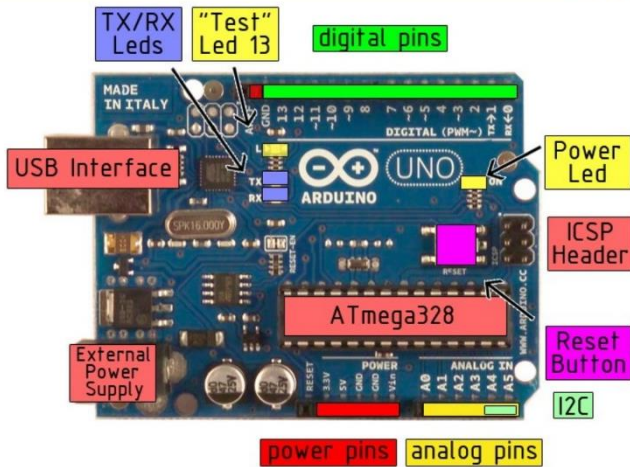


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

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



radiospares

RADIONICS



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. These 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.



radiospares

RADIONICS



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 it is 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).



radiospares

RADIONICS



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.



radiospares

RADIONICS



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-platform 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

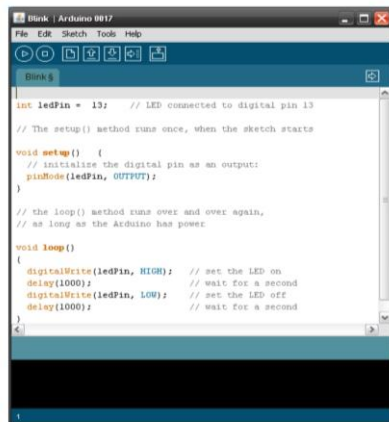
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**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.



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



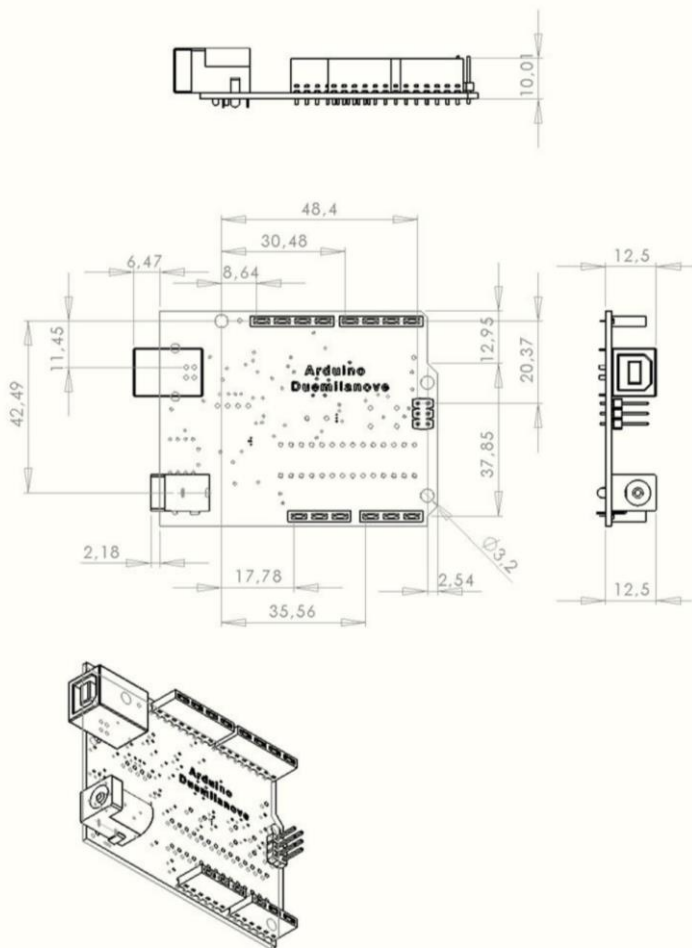
Blinking Led!



radiospares

RADIONICS





radiospares

RADIONICS



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.



Environmental 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.



radiospares

RADIONICS



Data Sheet, Rev. 1.1, December 2004

BTS 7960

High Current PN Half Bridge
NovalithIC™

43 A, 7 mΩ + 9 mΩ

Automotive Power



Never stop thinking.



High Current PN Half Bridge NovalithIC™

BTS 7960B

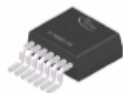
BTS 7960P

Product Summary

The **BTS 7960** is a fully integrated high current half bridge for motor drive applications. It is part of the **NovalithIC™** family containing one p-channel highside MOSFET and one n-channel lowside MOSFET with an integrated driver IC in one package. Due to the p-channel highside switch the need for a charge pump is eliminated thus minimizing EMI. Interfacing to a microcontroller is made easy by the integrated driver IC which features logic level inputs, diagnosis with current sense, slew rate adjustment, dead time generation and protection against overtemperature, overvoltage, undervoltage, overcurrent and short circuit.

The **BTS 7960** provides a cost optimized solution for protected high current PWM motor drives with very low board space consumption.

BTS 7960B
P-TO-263-7



BTS 7960P
P-TO-220-7



Basic Features

- Path resistance of typ. 16 mΩ @ 25 °C
- Low quiescent current of typ. 7 μA @ 25 °C
- PWM capability of up to 25 kHz combined with active freewheeling
- Switched mode current limitation for reduced power dissipation in overcurrent
- Current limitation level of 43 A typ.
- Status flag diagnosis with current sense capability
- Overtemperature shut down with latch behaviour
- Overvoltage lock out
- Undervoltage shut down
- Driver circuit with logic level inputs
- Adjustable slew rates for optimized EMI

Type	Ordering Code	Package
BTS 7960B	Q67060-S6160	P-TO-263-7
BTS 7960P	on request	P-TO-220-7

1 Overview

The **BTS 7960** is part of the **NovalithIC™** family containing three separate chips in one package: One p-channel highside MOSFET and one n-channel lowside MOSFET together with a driver IC, forming a fully integrated high current half-bridge. All three chips are mounted on one common leadframe, using the chip on chip and chip by chip technology. The power switches utilize vertical MOS technologies to ensure optimum on state resistance. Due to the p-channel highside switch the need for a charge pump is eliminated thus minimizing EMI. Interfacing to a microcontroller is made easy by the integrated driver IC which features logic level inputs, diagnosis with current sense, slew rate adjustment, dead time generation and protection against overtemperature, overvoltage, undervoltage, overcurrent and short circuit. The BTS 7960 can be combined with other BTS 7960 to form H-bridge and 3-phase drive configurations.

1.1 Block Diagram

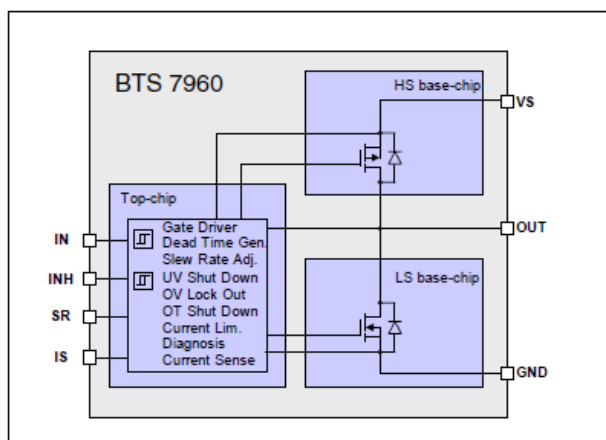


Figure 1 Block Diagram

1.2 Terms

Following figure shows the terms used in this data sheet.

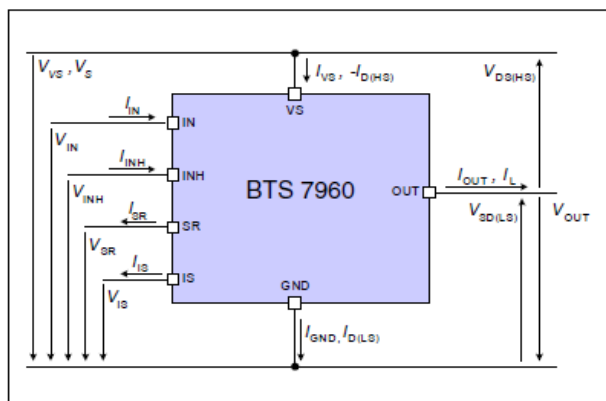


Figure 2 Terms

2 Pin Configuration

2.1 Pin Assignment

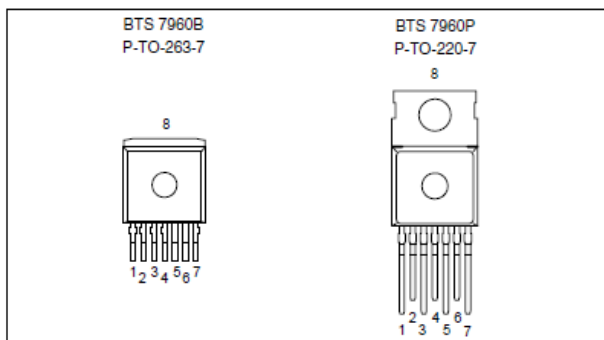


Figure 3 Pin Assignment BTS 7960B and BTS 7960P (top view)

2.2 Pin Definitions and Functions

Pin	Symbol	I/O	Function
1	GND	-	Ground
2	IN	I	Input Defines whether high- or lowside switch is activated
3	INH	I	Inhibit When set to low device goes in sleep mode
4,8	OUT	O	Power output of the bridge
5	SR	I	Slew Rate The slew rate of the power switches can be adjusted by connecting a resistor between SR and GND
6	IS	O	Current Sense and Diagnosis
7	VS	-	Supply

Bold type: pin needs power wiring

3 Maximum Ratings

-40 °C < T_J < 150 °C (unless otherwise specified)

Pos	Parameter	Symbol	Limits		Unit	Test Condition
			min	max		
Electrical Maximum Ratings						
3.0.1	Supply voltage	V_{VS}	-0.3	45	V	
3.0.2	Logic Input Voltage	V_{IN} V_{INH}	-0.3	5.3	V	
3.0.3	HS/LS continuous drain current	$I_{D(HS)}$ $I_{D(LS)}$	-40	40 ¹⁾	A	$T_C < 85^{\circ}\text{C}$ switch active
3.0.4	HS pulsed drain current	$I_{D(HS)}$	-60	60 ¹⁾	A	$T_C < 85^{\circ}\text{C}$
3.0.5	LS pulsed drain current	$I_{D(LS)}$	-60	60 ¹⁾	A	$t_{\text{pulse}} = 10\text{ms}$
3.0.6	Voltage at SR pin	V_{SR}	-0.3	1.0	V	
3.0.7	Voltage between VS and IS pin	$V_{VS} - V_{IS}$	-0.3	45	V	
3.0.8	Voltage at IS pin	V_{IS}	-20	45	V	
Thermal Maximum Ratings						
3.0.9	Junction temperature	T_J	-40	150	°C	
3.0.10	Storage temperature	T_{stg}	-55	150	°C	
ESD Susceptibility						
3.0.11	ESD susceptibility HBM	V_{ESD}			kV	according to EIA/ JESD 22-A 114B
	IN, INH, SR, IS OUT, GND, VS		-2 -6	2 6		

¹⁾ Maximum reachable current may be smaller depending on current limitation level

Note: Maximum ratings are absolute ratings; exceeding any one of these values may cause irreversible damage to the device. Exposure to maximum rating conditions for extended periods of time may affect device reliability

4 Block Description and Characteristics

4.1 Supply Characteristics

– 40 °C < T_J < 150 °C, 8 V < V_S < 18 V, I_L = 0 A (unless otherwise specified)

Pos.	Parameter	Symbol	Limit Values			Unit	Test Conditions
			min.	typ.	max.		
General							
4.1.1	Operating Voltage	V_S	5.5	–	27.5	V	
4.1.2	Supply Current	$I_{VS(on)}$	–	2	3	mA	$V_{INH} = 5\text{ V}$ $V_{IN} = 0\text{ V or } 5\text{ V}$ $R_{SP}=0\ \Omega$ DC-mode normal operation (no fault condition)
4.1.3	Quiescent Current	$I_{VS(off)}$	–	7	15	μA	$V_{INH} = 0\text{ V}$ $V_{IN} = 0\text{ V or } 5\text{ V}$ $T_J < 85\text{ }^{\circ}\text{C}$
			–	–	65	μA	$V_{INH} = 0\text{ V}$ $V_{IN} = 0\text{ V or } 5\text{ V}$

4.2 Power Stages

The power stages of the BTS 7960 consist of a p-channel vertical DMOS transistor for the high side switch and a n-channel vertical DMOS transistor for the low side switch. All protection and diagnostic functions are located in a separate top chip. Both switches can be operated up to 25 kHz, allowing active freewheeling and thus minimizing power dissipation in the forward operation of the integrated diodes.

The on state resistance R_{ON} is dependent on the supply voltage V_S as well as on the junction temperature T_J . The typical on state resistance characteristics are shown in Figure 4.

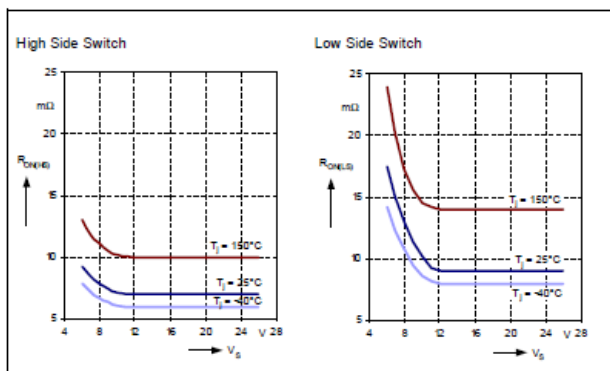


Figure 4 Typical On State Resistance vs. Supply Voltage

4.2.1 Power Stages - Static Characteristics

– 40 °C < T_J < 150 °C, 8 V < V_S < 18 V (unless otherwise specified)

Pos.	Parameter	Symbol	Limit Values			Unit	Test Conditions
			min.	typ.	max.		
High Side Switch - Static Characteristics							
4.2.1	On state high side resistance	$R_{ON(HS)}$	—	7 10	9 12.5	mΩ	$I_{OUT} = 9\text{ A}$ $V_S = 13.5\text{ V}$ $T_J = 25\text{ °C}$ $T_J = 150\text{ °C}$
4.2.2	Leakage current high side	$I_{L(LKHS)}$	—	—	1	μA	$V_{INH} = 0\text{ V}$ $V_{OUT} = 0\text{ V}$ $T_J < 85\text{ °C}$
			—	—	50	μA	$V_{INH} = 0\text{ V}$ $V_{OUT} = 0\text{ V}$ $T_J = 150\text{ °C}$
4.2.3	Reverse diode forward-voltage high side ¹⁾	$V_{DS(HS)}$	—	0.9 0.8 —	1.5 1.1 0.8	V	$I_{OUT} = -9\text{ A}$ $T_J = -40\text{ °C}$ $T_J = 25\text{ °C}$ $T_J = 150\text{ °C}$
Low Side Switch - Static Characteristics							
4.2.4	On state low side resistance	$R_{ON(LS)}$	—	9 14	12 18	mΩ	$I_{OUT} = -9\text{ A}$ $V_S = 13.5\text{ V}$ $T_J = 25\text{ °C}$ $T_J = 150\text{ °C}$
4.2.5	Leakage current low side	$I_{L(LKLS)}$	—	—	1	μA	$V_{INH} = 0\text{ V}$ $V_{OUT} = V_S$ $T_J < 85\text{ °C}$
			—	—	15	μA	$V_{INH} = 0\text{ V}$ $V_{OUT} = V_S$ $T_J = 150\text{ °C}$
4.2.6	Reverse diode forward-voltage low side ¹⁾	$V_{SD(LS)}$	—	0.9 0.8 —	1.5 1.1 0.8	V	$I_{OUT} = 9\text{ A}$ $T_J = -40\text{ °C}$ $T_J = 25\text{ °C}$ $T_J = 150\text{ °C}$

¹⁾ Due to active freewheeling, diode is conducting only for a few μs, depending on R_{SR}

4.2.2 Switching Times

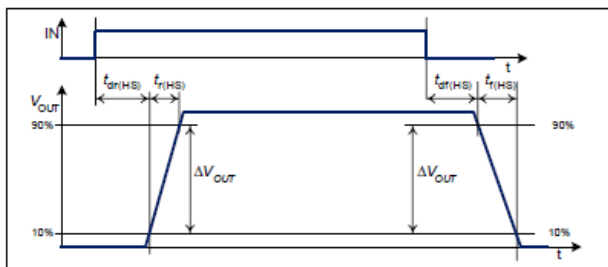


Figure 5 Definition of switching times high side (R_{load} to GND)

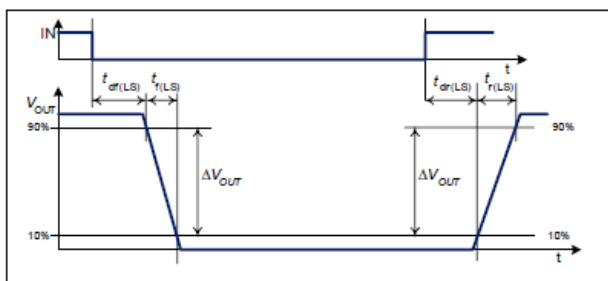


Figure 6 Definition of switching times low side (R_{load} to VS)

Due to the timing differences for the rising and the falling edge there will be a slight difference between the length of the input pulse and the length of the output pulse. It can be calculated using the following formulas:

- $\Delta t_{HS} = (t_{dr(HS)} + 0.5 t_{rl(HS)}) - (t_{dr(HS)} + 0.5 t_{rl(HS)})$
- $\Delta t_{LS} = (t_{dr(LS)} + 0.5 t_{rl(LS)}) - (t_{dr(LS)} + 0.5 t_{rl(LS)})$

DAFTAR RIWAYAT HIDUP



Nama : Sofyan A Rahman
TTL : Surabaya, 21 Februari 1997
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Bogen I/28c Surabaya
Telp/Hp : 081358699548
E-mail : sofyanarhmn11@gmail.com

RIWAYAT PENDIDIKAN

1. 2003-2009 : SDN Rangkah VIII Surabaya
2. 2009-2012 : SMPN 3 Surabaya
3. 2012-2015 : SMAN 6 Surabaya
4. 2015-2019 : Departemen Teknik Elektro Otomasi,
Program Studi Komputer Kontrol, FV
Institut Teknologi Sepuluh Nopember
(ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. PAL Indonesia (Persero)

PENGALAMAN ORGANISASI

1. Staff SC LEGO NXT IARC 2016
2. Staff SC LEGO NXT IARC 2017

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