

PERANCANGAN ALAT PENCUCI DAN PENGERING TANGAN OTOMATIS MENGGUNAKAN MIKROKONTROLLER ATMEGA16 DAN SCROLLING TEXT MESSAGE DISPLAY

Desy Kristyawati^[1], Ilham Nurcahyo^[2]
Teknik Elektro, Fakultas Teknologi Industri, Universitas Gunadarma
Jl. Margonda Raya 100 Depok 16424

Abstrak

Pada saat ini, mencuci tangan masih menggunakan prinsip manual dalam penggunaannya yaitu, mencuci tangan masih menggunakan air keran, dan mengeringkan tangan dengan handuk kecil. Oleh karena itu dibuatlah alat pencuci dan pengering tangan otomatis, Seiring dengan perkembangan teknologi semakin canggih, pembuatan alat ini dapat menggunakan sebuah mikrokontroler, dengan memanfaatkan fungsionalitas dari mikrokontroler ini kemudian dibuat sebuah mesin cuci dan pengering berbasis mikrokontroler ATMEGA16, menggunakan sensor LDR sebagai pendeteksi tangan, Relay sebagai pengendali pompa atau kipas, dan DOTMATRIX sebagai display. Dimana keuntungan dari alat ini dibandingkan dengan alat yang ada bahwa mesin cuci dan pengering tangan dapat bekerja lebih efisien dan dengan cepat adalah karena penggunaan sistem otomatis.

Kata Kunci : Sensor, ATMega16, Relay.

PENDAHULUAN

Latar Belakang Masalah

Di dalam restoran dan di kantor ataupun di pusat keramaian seperti mall, proses pencucian tangan masih banyak dilakukan secara manual sangatlah tidak efisien, dan membutuhkan waktu yang relatif lama. Sebenarnya jika proses pelayanan tersebut dapat diotomatisasikan akan sangat menguntungkan, baik itu bagi perusahaan yang bersangkutan maupun bagi pengguna itu sendiri. Maka dibuatlah alat otomasi pencuci dan pengering tangan menggunakan mikrokontroler seri ATMEGA16 dan LDR.

Komponen yang digunakan dalam pembuatan alat sistem otomatisasi pencuci dan pengering tangan sangat banyak dipasaran. Sebagian besar komponen berasal dari bahan semi konduktor. Simulasi sistem otomatisasi pencuci dan pengering tangan menggunakan LDR sebagai input utama untuk menentukan posisi tangan sehingga air bisa mengalir dan pengering tangan dapat menyala. Mikrokontroler dalam simulasi sistem otomatisasi pencuci dan pengering tangan sebagai unit pemroses data, dengan output berupa relay untuk menggerakkan pompa dan

kipas, dan Display Dot Matriks sebagai tampilan.

Cara kerja alat ini jika sensor LDR pada pencuci tangan terhalang maka logika sensor diproses oleh mikrokontroler dan mengaktifkan relay yang berfungsi untuk menyalakan pompa, sehingga air dapat mengalir. Ketika sensor tidak terhalang maka keadaan relay menjadi off, sehingga pompa tidak menyala. Jika sensor LDR pada pengering tangan terhalang maka logika sensor diproses oleh mikrokontroler dan mengaktifkan relay yang berfungsi untuk menyalakan kipas pengering.

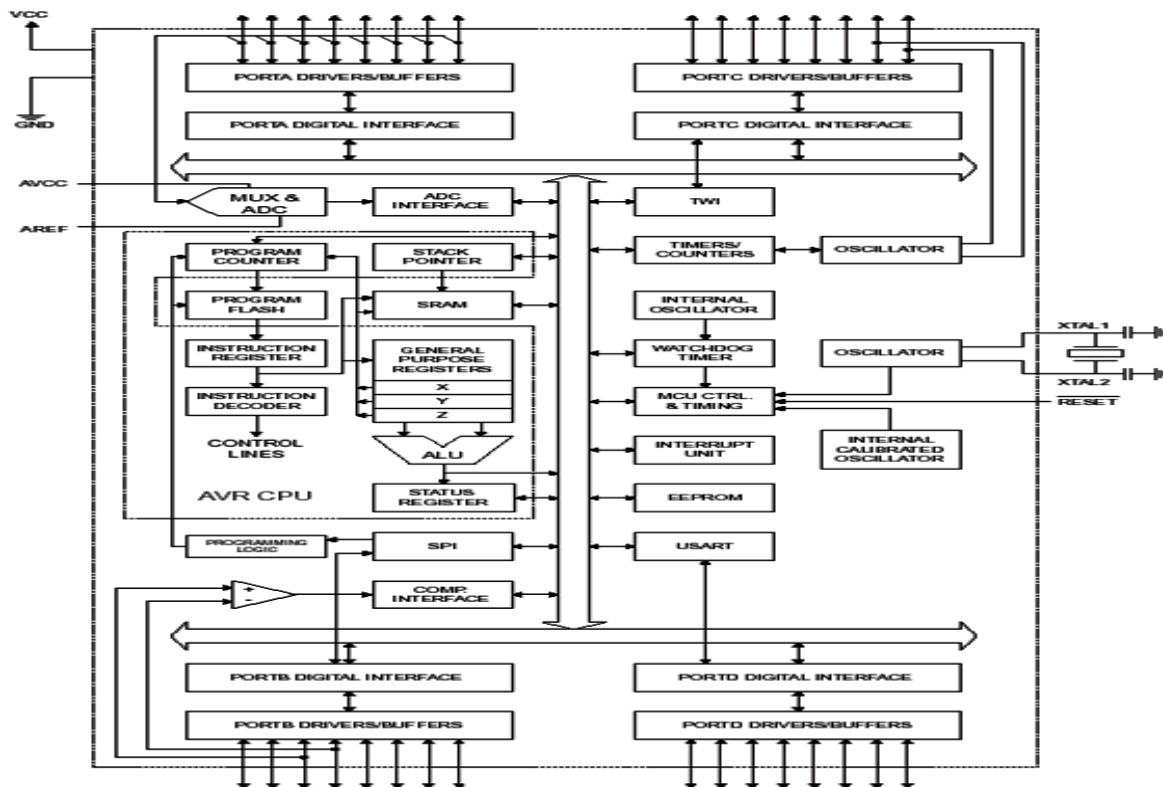
LANDASAN TEORI

Mikrokontroler ATMega16

Mikrokontroler AVR adalah mikro-kontroler RISC 8 bit berdasarkan arsitektur *Harvard*, yang dibuat oleh Atmel tahun 1996. AVR memiliki keunggulan dibandingkan dengan mikrokontroler lain, keunggulan AVR yaitu AVR memiliki kecepatan eksekusi program yang lebih cepat, karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock*, lebih cepat dibandingkan MCS51 yang

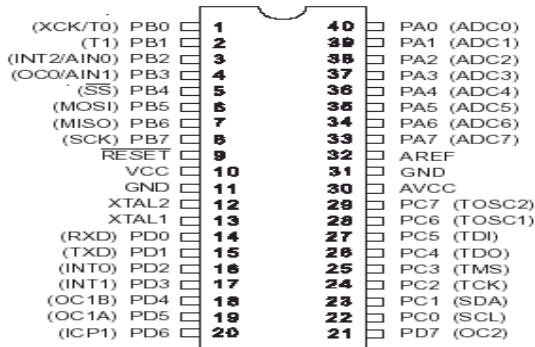
membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi. Mikrokontroler ATmega16 memiliki fitur yang lengkap (ADC internal, EEPROM internal, *Timer/Counter*, *Watchdog Timer*, PWM, Port I/O, komunikasi serial, Komparator, I2C,dll). Berikut ini merupakan beberapa spesifikasi ATmega16:

1. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16 Mhz.
2. Memiliki kapasitas *flash* memori 16Kbyte, EEPROM 512 Byte, dan SRAM 1Kbyte.
3. Saluran Port I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
4. CPU yang terdiri atas 32 buah register.
5. User interupsi internal dan eksternal.
6. Port USART sebagai komunikasi serial.
7. Konsumsi daya rendah (DC 5V).
8. Fitur *peripheral*, yang terdiri dari :
 - a. Tiga buah *Timer/Counter* dengan perbandingan
 - 2 (dua) buah *Timer/Counter* 8 bit dengan *Prescaler* terpisah dan *Mode Compare*
 - 1 (satu) buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, *Mode Compare*, dan *Mode Capture*
 - b. *Real Time Counter* dengan osilator tersendiri
 - c. *channel* PWM
 - d. 8 *channel*, 10-bit ADC
 - 8 *Single-ended Channel*
 - 7 *Differential Channel* hanya pada kemasan TQFP
 - 2 *Differential Channel* dengan *Programmable Gain* 1x, 10x, atau 200x
 - e. *Byte-oriented Two-wire Serial Interface*
 - f. Antarmuka SPI
 - g. *Watchdog Timer* dengan osilator internal
 - h. *On-chip Analog Comparator*



Gambar 1. Blok Diagram ATmega16

Konfigurasi Pin ATmega16



Gambar 2. Konfigurasi Pin ATmega16

Konfigurasi pin ATmega16 dengan kemasan 40 pin DIP (*Dual In-line Package*) dapat dilihat pada Gambar 2.2. Dari gambar di atas dapat dijelaskan fungsi dari masing-masing pin ATmega16 sebagai berikut:

1. Vcc merupakan pin yang berfungsi sebagai masukan catu daya.
2. GND merupakan pin *Ground*
3. Port A (PA0...7) merupakan pin *input/output* dua arah dan pin masukan ADC.
4. Port B (PB0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus seperti SPI, MISO, MOSI, SS, AIN1/OC0, AIN0/INT2, T1, T0 T1/XCK
5. Port C (PC0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus, seperti TOSC2, TOSC1, TDI, TDO, TMS, TCK, SDA, SCL
6. Port D (PD0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus, seperti RXD, TXD, INT0, INT1, OC1B, OC1A, ICP1
7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal
9. AVCC merupakan pin masukan tegangan untuk ADC
10. AREF merupakan pin masukan tegangan referensi ADC

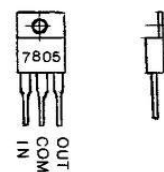
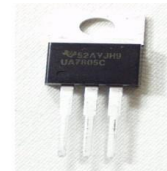
Register File	Data Address
R0	\$0000
R1	\$0001
R2	\$0002
...	...
R29	\$001D
R30	\$001E
R31	\$001F
I/O Registers	
\$00	\$0020
\$01	\$0021
\$02	\$0022
...	...
\$3D	\$005D
\$3E	\$005E
\$3F	\$005F
Internal SRAM	
	\$0060
	\$0061
	...
	\$045E
	\$045F

Gambar 3. Peta Memori Data ATMEGA16

Memori Data EEPROM

ATmega16 terdiri dari 512 byte memori data EEPROM 8 bit, data dapat ditulis/dibaca dari memori ini, ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM mulai dari \$000 sampai \$1FF.

IC Regulator IC LM7805

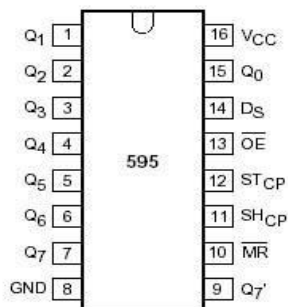


Gambar 4. IC LM7805

LM7805 merupakan IC pengatur tegangan (IC Regulator) yang menghasilkan tegangan keluaran 5 Volt DC dengan arus pada keluaran dapat mencapai 1 A.

IC 74HC595

IC 74HC595 (8-bit serial-in/ serial or parallel-output shift register) ini memiliki 8-bit input serial dengan 8-bit output serial atau output paralel dan IC ini juga memiliki *storage register* yang mana mempunyai pin input pulsa clock yang terpisah dengan shift registernya.



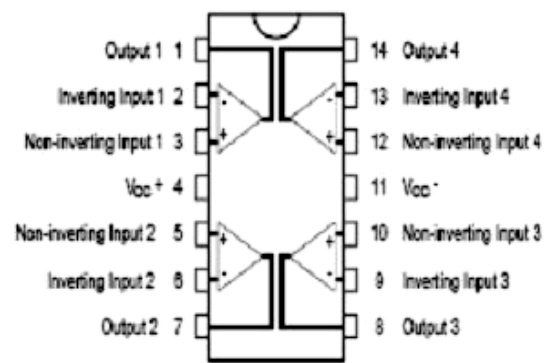
Gambar 5. Konfigurasi Pin IC 74HC595

Tabel 1. Keterangan Pin IC 74HC595

Symbol	Pin	Description
Q1	1	parallel data output 1
Q2	2	parallel data output 2
Q3	3	parallel data output 3
Q4	4	parallel data output 4
Q5	5	parallel data output 5
Q6	6	parallel data output 6
Q7	7	parallel data output 7
GND	8	ground (0 V)
Q7S	9	serial data output
MR	10	master reset (active LOW)
SHCP	11	shift register clock input
STCP	12	storage register clock input
OE	13	output enable input (active LOW)
DS	14	serial data input
Q0	15	parallel data output 0
V _{CC}	16	supply voltage

IC LM 324

IC LM324 merupakan IC Operational Amplifier, IC ini mempunyai 4 buah op-amp yang berfungsi sebagai comparator. IC ini mempunyai tegangan kerja antara +5 V sampai +15V untuk +V_{CC} dan -5V sampai -15V untuk -V_{CC}. Adapun definisi dari masing-masing pin IC LM324 adalah sebagai berikut :



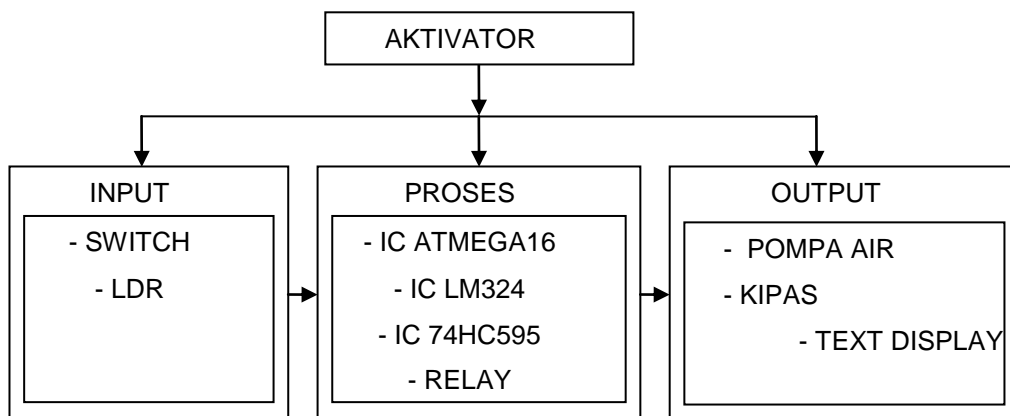
Gambar 7. Konfigurasi Pin IC LM324

- pin 1,7,8,14 (Output)
Merupakan sinyal output.
- Pin 2,6,9,13 (Inverting Input)
Semua sinyal input yang berada di pin ini akan mempunyai output yang berkebalikan dari input.
- Pin 3,5,10,12 (Non-inverting input)
Semua sinyal input yang berada di pin ini akan mempunyai output yang sama dengan input (tidak berkebalikan).
- Pin 4 (+V_{CC})
Pin ini dapat beroperasi pada tegangan antara +5 Volt sampai +15 Volt.
- Pin 11 (-V_{CC})
Pin ini dapat beroperasi pada tegangan antara -5 Volt sampai -15 Volt.

PERANCANGAN ALAT

Blok Diagram

Secara umum blok diagram Alat Pencuci Dan Pengering Tangan Otomatis Menggunakan ATMEGA16 Dan Scrolling Text Message sebagai berikut :

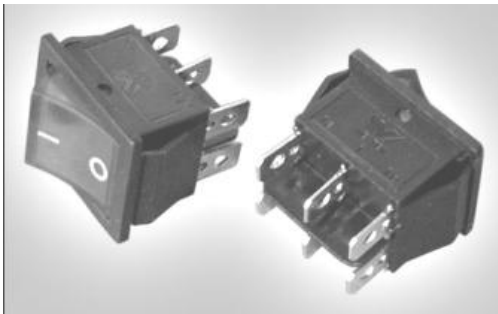


Gambar 9. Blok Diagram Alat Pencuci Dan Pengering Tangan Otomatis Menggunakan ATMEGA16 Dan Scrolling Text Message.

Blok Aktivator

Pertama sumber tegangan DC dapat berupa baterai atau adaptor. Alat pencuci dan pengeringan menggunakan sumber tegangan DC dari adaptor.

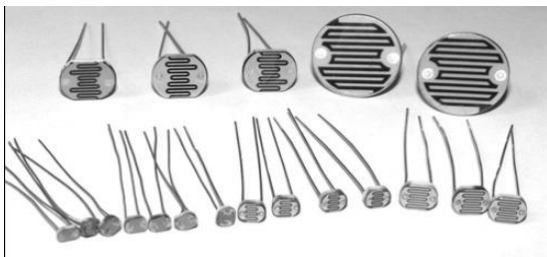
Blok input Switch



Gambar 10. switch

Pada saat kabel power sudah terhubung ke sumber tegangan, maka arus akan mengalir melalui Switch, jika switch dalam keadaan ON maka supply tegangan akan mengalir ke dalam rangkaian, dan sebaliknya jika switch dalam keadaan OFF maka supply tegangan tidak ada yang mengalir.

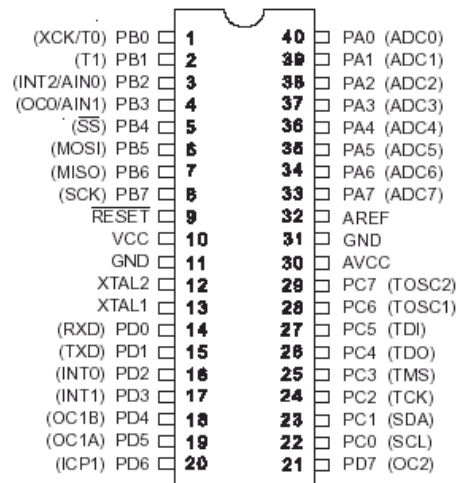
LDR



Gambar 11. Light Dependent Resistor (LDR)

Saat supply tegangan mengalir ke dalam rangkaian, kondisi awal sensor berlogika 1 (high), karena sensor LDR mendapat cahaya dari LED dan resistansi / hambatan LDR menjadi kecil. Ketika sensor mendeteksi keberadaan tangan maka kondisi sensor berlogika 0 (low), karena cahaya LED terhalang oleh tangan dan LDR tidak mendapat cahaya sehingga resistansi LDR menjadi besar. Output dari sensor kemudian akan diproses oleh mikrokontroller ATMEGA 16 untuk mengendalikan rangkaian driver pompa dan kipas.

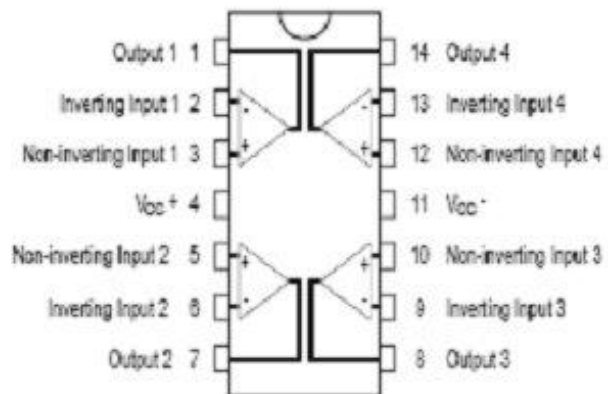
Blok Proses IC ATMEGA16



Gambar 12. IC ATMEGA16

IC ATMEGA16 digunakan untuk mengontrol driver pompa dan kipas sesuai kondisi pada sensor. Jika sensor 1 berlogika 0 (low) maka mikrokontroller akan memberikan logika 1 (high) ke dalam rangkaian driver dan mengaktifkan relay 1 sebagai saklar untuk mengaktifkan pompa sampai kondisi sensor berlogika 1 (high) kembali. Jika sensor 2 berlogika 0 (low) maka mikrokontroller akan memberikan logika 1 (high) ke dalam rangkaian driver dan mengaktifkan relay 2 sebagai saklar untuk mengaktifkan kipas sampai kondisi sensor berlogika 1 (high) kembali. Agar dapat berjalan sesuai algoritma tersebut, maka di buatlah program dan di simpan dalam memory flash IC ATMEGA 16.

IC LM324

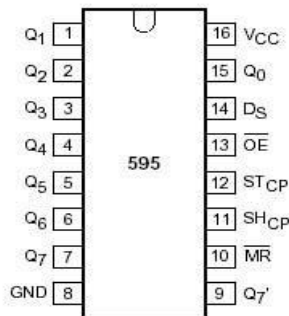


Gambar 13. IC LM324

Sensor LDR adalah sensor yang resistansinya berubah terhadap cahaya, semakin terang cahaya maka semakin kecil resistansinya, begitu juga sebaliknya jika semakin redup cahaya yang di dapat maka semakin besar resistansinya. Output sensor LDR terhubung ke kaki inverting IC LM324 yang berfungsi sebagai komparator / pembanding anatar tegangan sensor dan tegangan pada trimpot. Trimpot di gunakan untuk mengatur sensitivitas sensor LDR. Adapun rumus komparator/pembanding yang digunakan sebagai berikut:

$$V_{OUT} = (V_b - V_a) \pm 90\% V_{CC}$$

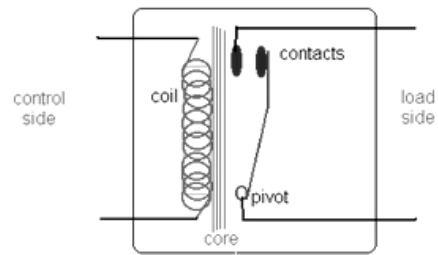
IC 74HC595



Gambar 14. IC 74HC595

IC 74HC595 yang memiliki paralel output untuk menentukan pergeseran bit yang digunakan untuk mengendalikan kolom dotmatrix, karena IC ini mempunyai shift register 8 tingkat untuk mengatur/ mengubah input data serial menjadi output data paralel. IC ini selain memiliki shift register juga memiliki storage register untuk menyimpan hasil atau data yang didapat dari shift registernya yang mana input pulsa clock untuk storage register terpisah dengan input clock bagi shift registernya. Input data serial berasal dari pin 14 (DS) yang kemudian masuk ke shift register. Untuk pulsa clock pemicu inputan shift register berasal dari pin 11 (SH_{CP}). Kemudian data paralel yang dihasilkan oleh shift register dikirimkan ke storage register dengan pulsa clock pemicunya berasal dari pin 12 (ST_{CP}). Lalu data yang telah disimpan pada storage register akan dikeluarkan pada pin Q₀ sampai Q₇ bila pada pin 13 (OE') diberikan kondisi masukan rendah (LOW).

RELAY



INSIDE A SPST RELAY pic r-1a

Gambar 15. Prinsip Kerja Relay

Relay terdiri dari Coil & Contact coil adalah gulungan kawat yang mendapat arus listrik, sedang contact adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik dicoil. Contact ada 2 jenis : Normally Open (kondisi awal sebelum diaktifkan open), dan Normally Closed (kondisi awal sebelum diaktifkan close). Secara sederhana berikut ini prinsip kerja dari relay : ketika Coil mendapat energi listrik (energized), akan timbul gaya elektromagnet yang akan menarik armature yang berpegas, dan contact akan menutup

Blok Output POMPA AIR



Gambar 16. Pompa Air

Saat relay 1 dalam kondisi aktif maka POMPA AIR mendapat sumber tegangan dan pompa dapat berputar mengalirkan air saat keberadaan tangan terdeteksi.

KIPAS



Gambar 17. Kipas Pengering

Saat relay 2 dalam kondisi aktif maka KIPAS mendapat sumber tegangan dan kipas dapat berputar mengalirkan udara yang berguna sebagai pengerig tangan.

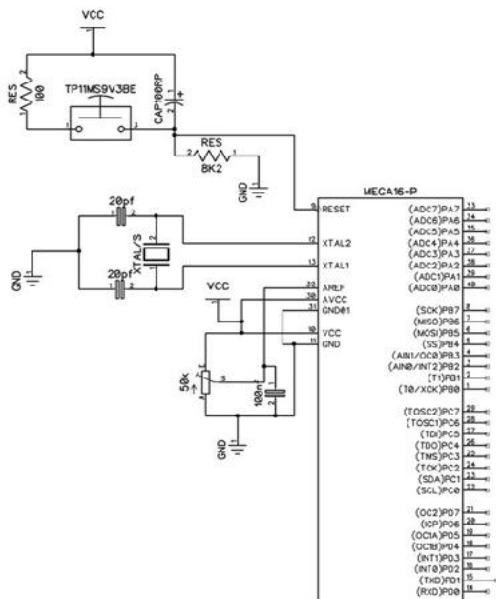
TEXT DISPLAY



Gambar 18. Dotmatrix 5 X 7

TEXT DISPLAY yang terdiri dari dotmatrix 5 x 7 sebanyak 6 buah berguna untuk menampilkan pesan – pesan singkat pada saat tidak ada perubahan sensor (standby) yang mana pesan tersebut berjalan dari arah kanan ke kiri.

BLOK RANGKAIAN BLOK MIKROKONTROLLER ATMEGA16

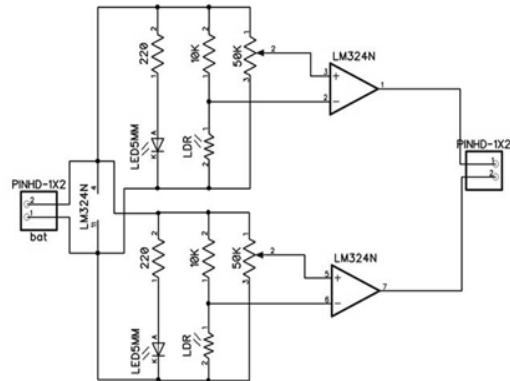


Gambar 19. Mikrokontroller ATMEGA16

IC mikrokontroller digunakan sebagai pemroses dan pengendali karena IC ini bekerja sesuai program yang diisikan oleh user ke dalam mikrokontroller. Input dari sensor yang terhubung pada kaki PORTD.2 (INT0) akan diproses dan mengeluarkan output pada PORTA.0 dan menyalakan pompa air. Input dari sensor yang terhubung pada kaki PORTD.3 (INT1) akan diproses dan

mengeluarkan output pada PORTA.1 dan menyalakan kipas. PORTB digunakan untuk mengendalikan baris pada dotmatrix, dan PORTC terhubung pada IC74HC595 sebagai pengendali kolom dotmatrix.

BLOK SENSOR

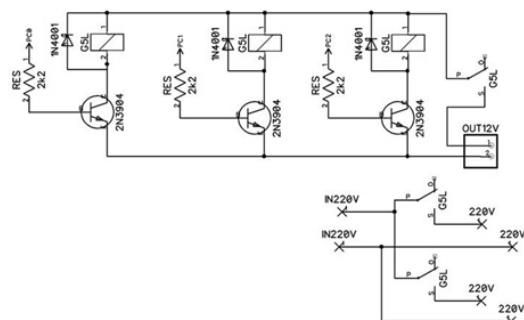


Gambar 20. blok sensor

Rangkaian sensor ini menggunakan sensor LDR. Sensor LDR adalah sensor yang resistansinya berubah terhadap cahaya, semakin terang cahaya maka semakin kecil resistansinya, begitu juga sebaliknya jika semakin redup cahaya yang di dapat maka semakin besar resistansinya. Output sensor LDR terhubung ke kaki inverting IC LM324 yang berfungsi sebagai komparator / pembanding anatar tegangan sensor dan tegangan pada trimpot. Trimpot di gunakan untuk mengatur sensitivitas sensor LDR. Adapun rumus komparator/pembanding yang diguna-kan sebagai berikut:

$$V_{OUT} = (V_b - V_a) \pm 90\% VCC$$

BLOK RELAY

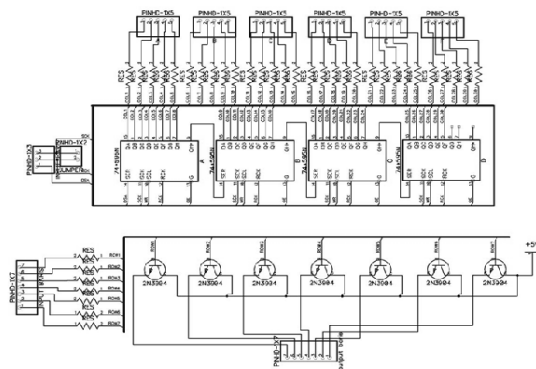


Gambar 21. Blok rangkaian relay

Pada rangkaian relay ini menggunakan relay magnetis 12v dan transistor 2n3904 sebagai pengendali relay. Jika kaki basis pada transistor 2n3904 diberi logika 1 maka

transistor menjadi saturasi sehingga arus mengalir dari kaki kolektor ke kaki emitor dan relay menjadi aktif dari keadaan NC (Normaly Close) menjadi NO (Normali Open).

BLOK DRIVER DOT MATRIX



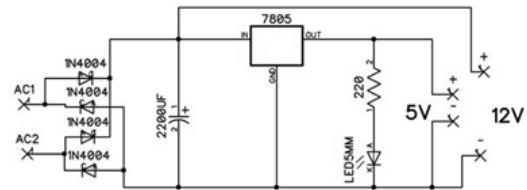
Gambar 22. Rangkaian Driver Dot Matrix

IC 74HC595 yang memiliki paralel output untuk menentukan pergeseran bit yang digunakan untuk mengendalikan kolom dotmatrix, karena IC ini mempunyai shift register 8 tingkat untuk mengatur/ mengubah input data serial menjadi output data paralel. IC ini selain memiliki shift register juga memiliki storage register untuk menyimpan hasil atau data yang didapat dari shift registernya yang mana input pulsa clock untuk storage register terpisah dengan input clock bagi shift registernya. Input data serial berasal dari pin 14 (DS) yang kemudian masuk ke shift register. Untuk pulsa clock pemicu inputan shift register berasal dari pin 11 (SH_{CP}). Kemudian data paralel yang dihasilkan oleh shift register dikirimkan ke storage register dengan pulsa clock pemicunya berasal dari pin 12 (ST_{CP}). Lalu data yang telah disimpan pada storage register akan dikeluarkan pada pin Q₀ sampai Q₇ bila pada pin 13 (OE') diberikan kondisi masukan rendah (LOW).

Rangkaian driver dot matrix terdiri rangkaian pengendali kolom dan rangkaian pengendali baris. Pada rangkaian pengendali kolom menggunakan IC 74HC595. IC 74HC595 berfungsi sebagai shift register untuk mengendalikan kolom dotmatrix dengan input serial dan output paralel. Jika pin serial di beri logika 1 kemudian pin SHCP STCP di klok sebanyak 1 kali maka pada pin D0 akan berlogika high, kemudian jika pin SERIAL di beri logika 0 kemudian pin SHCP dan STCP di klok sebanyak N kolom, maka output pada D0 akan bergeser sebanyak N kolom. Pada

rangkaian pengendali baris menggunakan transistor NPN 2n2222

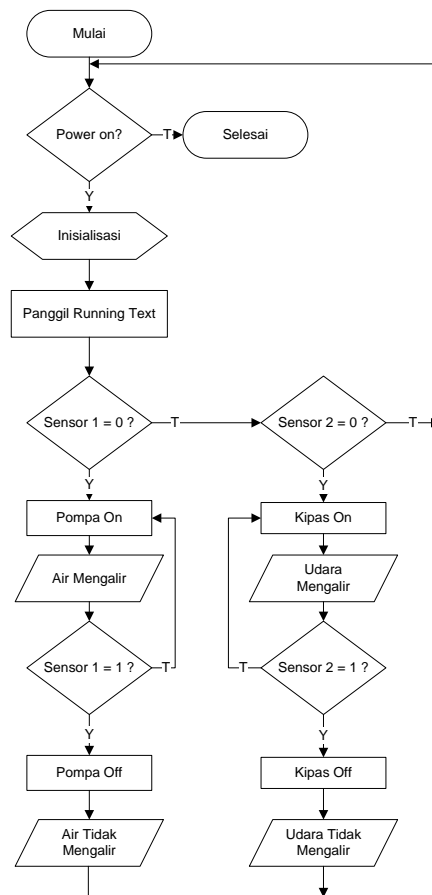
BLOK CATU DAYA



Gambar 23 Blok Catu Daya

Blok catu daya dari tegangan AC 220 diturunkan oleh trafo stepdown K3 menjadi 15V yang kemudian masuk ke dalam dioda bridge untuk dijadikan tegangan DC. Sebagian tegangan tersebut dihubungkan ke IC regulator LM7805 untuk mendapatkan tegangan sebesar 5V sebagai sumber tegangan untuk mengaktifkan keseluruhan IC yang ada pada rangkaian (VCC) yang mana sebagai indikatornya adalah LED.

ANALISA FLOWCHART



Gambar 24 Flowchart

Analisa Flowchart

- Langkah pertama dimulai dari terminal start dan kemudian tentukan apakah sumber tegangan aktif (Power ON) jika belum maka akhiri program, jika ya maka ke langkah berikutnya.
- Langkah kedua yaitu proses untuk menampilkan message / pesan running text pada dotmatrix 30 x7.
- Langkah ketiga mengambil input dari sensor 1. apakah kondisi sensor 1 berlogika 0 (low), jika ya maka pompa air menyala hingga kondisi sensor 1 berlogika 1 (high) dan kembali ke proses running text.
- Langkah keempat mengambil input dari sensor 2. apakah kondisi sensor 2 berlogika 0 (low), jika ya maka kipas menyala hingga kondisi sensor 2 berlogika 1 (high). kembali ke proses running text.
- Langkah kelima jika kondisi sensor 1 dan sensor 2 sama dengan 1 (high), maka lompat ke proses running text.

```
'=====Header=====
$regfile = "m16def.dat"
    'inisialisasi chip yang digunakan
$crystal = 8000000
    'inisialisasi Kristal
*****Inisialisasi Interrupt*****
Enable Interrupts
mengaktifkan semua interupsi
Enable Int0
    'mengaktifkan interupsi eksternal 0
Enable Int1
    'mengaktifkan interupsi eksternal 1
    On Int0 Pompa 'jika terjadi interupsi
    external 0, maka program akan
    melompat ke label pompa.
On Int1 Kipas 'jika terjadi interupsi external 1,
maka program akan melompat ke label kipas.

*****Pendeklarasian Variabel*****
Dim Str_temp As String * 1 , Reeks(250) As
Byte , Str_text As String * 40
Dim Char As Byte , C As Byte , Digit As Byte , L As
Byte , I As Byte
Dim Str_len As Byte , Col_len As Word , R As Byte
Dim X As Word , Y As Integer , Z As Byte
Dim Scroll_speed As Byte

*****Pendeklarasian Sub – Sub Program*****
Declare Sub Show_text
Declare Sub Show_text1
Declare Sub Set_text
Declare Sub Test_display
Declare Sub Clear_display
```

```
*****Konfigurasi Port Yang Digunakan*****
    Config Portc = Output
    'shift register 74HC595
    Config Portb = Output
    'Dotmatrix 5x7
    Config Portd = Input
    Config Porta = Output
*****Main Program*****
Call Test_display
Waitms 1
Main:
Z = 1
Gosub Text1
Gosub Text2
Gosub Text3
Gosub Text4
Goto Main
End
*****Sub Text 1*****
Text1:
Call Clear_display
Scroll_speed = Z
Str_text = " sudahkah anda mencuci tangan?"
Str_len = Len(str_text)
Col_len = Str_len * 6
Call Set_text
Call Show_text
Return
*****Sub Text 2*****
Text2:
Call Clear_display
Scroll_speed = Z
Str_text = " cucilah tangan setelah
beraktifitas"
Str_len = Len(str_text)
Col_len = Str_len * 6
Call Set_text
Call Show_text
Return
*****Sub Text 3*****
Text3:
Call Clear_display
Scroll_speed = Z
Str_text = " dan sebelum atau sesudah
makan"
Str_len = Len(str_text)
Col_len = Str_len * 6
Call Set_text
Call Show_text
Return
*****Sub Text 4*****
Text4:
Call Clear_display
Scroll_speed = Z
Str_text = " jagalah kebersihan :)"
Str_len = Len(str_text)
Col_len = Str_len * 6
Call Set_text
```

```

Call Show_text
Return
*****Sub Set Text *****
'set_text
Sub Set_text
X = 1
C = 1
For Digit = 1 To Str_len
Str_temp = Mid(str_text , Digit , 1)
Char = Asc(str_temp)
X = Char * 5
For I = 1 To 5
Reeks(c) = Lookup(x , Text_data)
X = X + 1
C = C + 1
Next I
Reeks(c) = 0
C = C + 1
Next Digit
End Sub Set_text
***Sub Menampilkan Text Pada Dotmatrix***
'show text
Sub Show_text
If Str_len > 5 Then
For L = 0 To Col_len
C = 1
  For R = 1 To Scroll_speed
    Reset Portc.0
    Toggle Portc.1
    Toggle Portc.1
    Toggle Portc.2
    Toggle Portc.2
    Char = C + L
    Portb = Reeks(char)
    Waitms 1
    C = C + 1
    Set Portc.0
    For X = 1 To 32
      Toggle Portc.1
      Toggle Portc.1
      Toggle Portc.2
      Toggle Portc.2
      Char = C + L
      Portb = Reeks(char)
      Waitms 1
      C = C + 1
    Next X
  C = 1
Next R
Next L
Else
For L = 0 To Col_len
C = 1
  For R = 1 To Scroll_speed
    Reset Portc.0
    Toggle Portc.1
    Toggle Portc.1
    Toggle Portc.2
    Toggle Portc.2
    Toggle Portc.2
    Portb = Reeks(c)
    Waitms 1
    C = C + 1
    Set Portc.0
    For X = 1 To 32
      Toggle Portc.1
      Toggle Portc.1
      Toggle Portc.2
      Toggle Portc.2
      Portb = Reeks(c)
      Waitms 1
      C = C + 1
    Next X
  C = 1
Next R
Next L
End Sub Show_text
*****Sub Test Display*****
Sub Test_display
Portb = 255
For X = 1 To 30
  Reset Portc.0
  Toggle Portc.1
  Toggle Portc.1
  Toggle Portc.2
  Toggle Portc.2
Next X
End Sub Test_display
*****Sub Clear Display*****
Sub Clear_display
Str_text = "
Call Set_text
For L = 0 To 1
C = 1
  For R = 1 To 1
    Reset Portc.0
    Toggle Portc.1
    Toggle Portc.1
    Toggle Portc.2
    Toggle Portc.2
    Portb = Reeks(c)
    Waitms 1
    C = C + 1
    Set Portc.0
    For X = 1 To 32
      Toggle Portc.1
      Toggle Portc.1
      Toggle Portc.2
      Toggle Portc.2
      Portb = Reeks(c)
      Waitms 1
      C = C + 1
    Next X
  C = 1
Next R
Next L
Toggle Portc.2
Portb = Reeks(c)
Waitms 1
C = C + 1
Set Portc.0
For X = 1 To 32
  Toggle Portc.1
  Toggle Portc.1
  Toggle Portc.2
  Toggle Portc.2
  Portb = Reeks(c)
  Waitms 1
  C = C + 1
Next X
C = 1
Next R
Next L

```

```

End Sub Clear_display
Data 54 , 73 , 86 , 32 , 80      '&
Data 0 , 0 , 11 , 7 , 0         "
*****Sub Program Pompa*****
Data 0 , 28 , 34 , 65 , 0       '(
Pompa:                          Data 0 , 65 , 34 , 28 , 0       ')'
Do                                Data 42 , 28 , 127 , 28 , 42    '*
Portb = &H00                      Data 8 , 8 , 62 , 8 , 8        '+
Set Porta.7                       Data 0 , 88 , 56 , 0 , 0       ','
Loop Until Pind.2 = 1             Data 8 , 8 , 8 , 8 , 8        '-
Reset Porta.7                    Data 0 , 96 , 96 , 0 , 0       ':'
Return                            Data 32 , 16 , 8 , 4 , 2       '/'
**Sub Program Kipas*****
Data 62 , 81 , 73 , 69 , 62      '0
Kipas:                            Data 0 , 66 , 127 , 64 , 0     '1
Do                                Data 114 , 73 , 73 , 73 , 70   '2
Portb = &H00                      Data 34 , 65 , 73 , 73 , 54   '3
Set Porta.1                      Data 24 , 20 , 18 , 127 , 16  '4
Loop Until Pind.3 = 1            Data 39 , 69 , 69 , 69 , 57   '5
Reset Porta.1                   Data 60 , 74 , 73 , 73 , 48   '6
Return                            Data 1 , 113 , 9 , 5 , 3      '7
*****Data Karakter Dotmatrix *****
Data 54 , 73 , 73 , 73 , 54     '8
Text_data:                       Data 6 , 73 , 73 , 41 , 30    '9
Data 0 , 0 , 0 , 0 , 0          ':'
Data 0 , 0 , 0 , 0 , 0          ':'
Data 0 , 0 , 0 , 0 , 0          '<'
Data 0 , 0 , 0 , 0 , 0          '='
Data 0 , 0 , 0 , 0 , 0          '>'
Data 0 , 0 , 0 , 0 , 0          '?'
Data 0 , 0 , 0 , 0 , 0          '@'
Data 0 , 0 , 0 , 0 , 0          'A'
Data 0 , 0 , 0 , 0 , 0          'B'
Data 0 , 0 , 0 , 0 , 0          'C'
Data 0 , 0 , 0 , 0 , 0          'D'
Data 0 , 0 , 0 , 0 , 0          'E'
Data 0 , 0 , 0 , 0 , 0          'F'
Data 0 , 0 , 0 , 0 , 0          'G'
Data 0 , 0 , 0 , 0 , 0          'H'
Data 0 , 0 , 0 , 0 , 0          'I'
Data 0 , 0 , 0 , 0 , 0          'J'
Data 0 , 0 , 0 , 0 , 0          'K'
Data 0 , 0 , 0 , 0 , 0          'L'
Data 0 , 0 , 0 , 0 , 0          'M'
Data 0 , 0 , 0 , 0 , 0          'N'
Data 0 , 0 , 0 , 0 , 0          'O'
Data 0 , 0 , 0 , 0 , 0          'P'
Data 0 , 0 , 0 , 0 , 0          'Q'
Data 0 , 0 , 0 , 0 , 0          'R'
Data 0 , 0 , 0 , 0 , 0          'S'
Data 0 , 0 , 0 , 0 , 0          'T'
Data 0 , 0 , 0 , 0 , 0          'U'
Data 0 , 0 , 0 , 0 , 0          'V'
Data 0 , 0 , 0 , 0 , 0          'W'
Data 0 , 0 , 0 , 0 , 0          'X'
Data 0 , 0 , 0 , 0 , 0          'Y'
Data 0 , 0 , 0 , 0 , 0          'space'
Data 0 , 0 , 95 , 0 , 0         '!'
Data 0 , 7 , 0 , 7 , 0         '"'
Data 20 , 127 , 20 , 127 , 20   '#'
Data 36 , 42 , 127 , 42 , 18   '$'
Data 39 , 21 , 107 , 84 , 114  '%'
Data 97 , 81 , 73 , 69 , 67     'Z'
Data 0 , 127 , 65 , 65 , 0     '['
Data 2 , 4 , 8 , 16 , 32       '\'
Data 0 , 65 , 65 , 127 , 0     ']'
Data 4 , 2 , 1 , 2 , 4         '^'
Data 64 , 64 , 64 , 64 , 64     '_'

```

Data 0 , 0 , 7 , 11 , 0	"
Data 32 , 84 , 84 , 84 , 56	'a
Data 127 , 40 , 68 , 68 , 56	'b
Data 0 , 56 , 68 , 68 , 68	'c
Data 56 , 68 , 68 , 40 , 127	'd
Data 56 , 84 , 84 , 84 , 24	'e
Data 8 , 126 , 9 , 9 , 2	'f
Data 8 , 84 , 84 , 84 , 60	'g
Data 127 , 16 , 8 , 8 , 112	'h
Data 0 , 68 , 125 , 0 , 0	'i
Data 32 , 64 , 68 , 61 , 0	'j
Data 0 , 127 , 16 , 40 , 68	'k
Data 0 , 65 , 127 , 64 , 0	'l
Data 124 , 4 , 120 , 4 , 120	'm
Data 124 , 8 , 4 , 4 , 120	'n
Data 56 , 68 , 68 , 68 , 56	'o
Data 124 , 20 , 20 , 20 , 8	'p
Data 8 , 20 , 20 , 20 , 124	'q
Data 124 , 8 , 4 , 4 , 8	'r
Data 72 , 84 , 84 , 84 , 36	's
Data 0 , 4 , 63 , 68 , 68	't
Data 60 , 64 , 64 , 32 , 124	'u
Data 28 , 32 , 64 , 32 , 28	'v
Data 60 , 64 , 60 , 64 , 60	'w
Data 68 , 40 , 16 , 40 , 68	'x
Data 12 , 80 , 80 , 80 , 60	'y
Data 68 , 100 , 84 , 76 , 68	'z
Data 0 , 8 , 54 , 65 , 0	{
Data 0 , 0 , 119 , 0 , 0	
Data 0 , 65 , 54 , 8 , 0	}
Data 2 , 1 , 2 , 4 , 2	~
Data 42 , 85 , 42 , 85 , 42	'

KESIMPULAN

Kesimpulan dari perancangan alat pencuci tangan dan penering tangan menggunakan mikrokontroler ATMEGA16 sebagai pengolah data, LDR sebagai sensor keberadaan tangan, RELAY sebagai aktuator, dan DOTMATRIX 30 x 7 sebagai display dapat berjalan sesuai dengan fungsi yang diinginkan. Mikrokontoller yang digunakan yaitu Mikrokontroller Keluarga AVR seri ATMEGA16, dimana mikrokontroller ini mempunyai kapasitas memory flash sebesar 16KB lebih besar dibandingkan Mikrokontroller Keluarga MCS51 seri AT89S51 yang mempunyai memory flash sebesar 4KB dan keunggulan AVR lainnya yaitu AVR memiliki kecepatan eksekusi program yang lebih cepat, karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock*, lebih cepat dibandingkan MCS51 yang membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi.

DAFTAR PUSTAKA

1. ATmega16 Preliminary Datasheet, Rev. F. 2002, Atmel Corporation
2. Fairchild Semiconductor, "MC78XX/LM78XX/MC78XXA, 3-Terminal 1A Positive Voltage Regulator", Datasheet, 2001.
<http://www.fairchildsemi.com/ds/LM/LM7805.pdf>
3. Introduction-proteus
<http://yd1chs.files.wordpress.com/2011/04/02-introduction-proteus.pdf>
4. Iswanto, "Design dan Implementasi Sistem Embeded Mikrokontroller ATmega8535 dengan Bahasa Basic", GAVA MEDIA, Yogyakarta, 2008
5. Mengenal bahasa BASIC pada BASCOM AVR
<http://fahmizaleeits.wordpress.com/2010/04/09/mengenal-bahasa-basic-pada-bascom-avr/>
6. Philips Semiconductors, "74HC/ HCT595 8-bit serial-in/serial or parallel-out shift register with output latches; 3-state", Datasheet, 4 Juni 1998.
<http://workshop.ee.itb.ac.id/upload/datasheet/74hc595.pdf>
7. Setiawan, Afrie, "Aplikasi Mikrokontrol-ler ATmega16 Menggunakan Basvom- AVR", Andi, Yogyakarta, 2011
8. Tutorial Software Bascom AVR
<http://fahmizaleeits.wordpress.com/2011/02/26/tutorial-software-bascom-avr/>
9. Wahyudin, Didin, "Belajar Mudah Mikrokontroller AT89s52 dengan bahasa basic menggunakan Bascom-8051", Andi, Yogyakarta, 2007
10. Wiyono, Didik, "Panduan Praktis Mikrokontroller Keluarga AVR Menggunakan DT-COMBO AVR-51 STARTER KIT DAN DT-COMBO AVR EXERCISE KIT", Innovative Electronics, 2007
11. "2N3904/MMBT3904/MMPQ3904/PZT3904", National Semiconductor, Datasheet 06/06/2012.
<http://www.makershed.com/v/vspfiles/asset/s/images/2n3904.pdf>
12. "2N3906/MMBT3906/MMPQ3906/PZT3906", National Semiconductor, Datasheet 06/06/2012.
<http://www.makershed.com/v/vspfiles/asset/s/images/2n3906.pdf>
13. "LM324/LM/DS", National Semiconductor, Datasheet 06/06/2012.
<https://www.national.com/ds/LM/LM124.pdf>