

DataSheet Motor DC gearbox



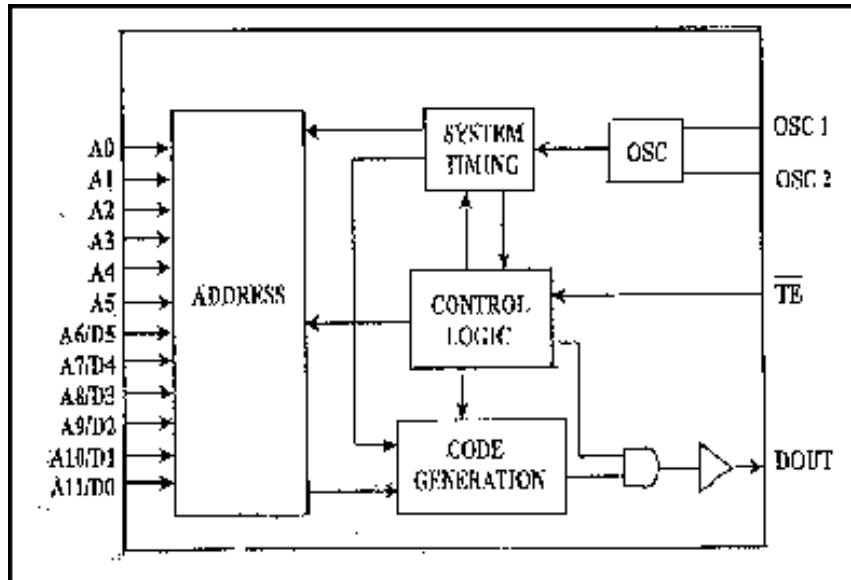
Spesifikasi Motor DC gearbox :

- Daya : 150 W
- Tegangan : 5V, 12V dan 24V DC
- Speed(ns) : 330 rpm
- Torsi(Nm) : 4.8 newtonmeter
- Gear : Free Gear

DataSheet Remote PT2262/2272

Operasi Sirkuit

Bagian Pengkodean PT2262



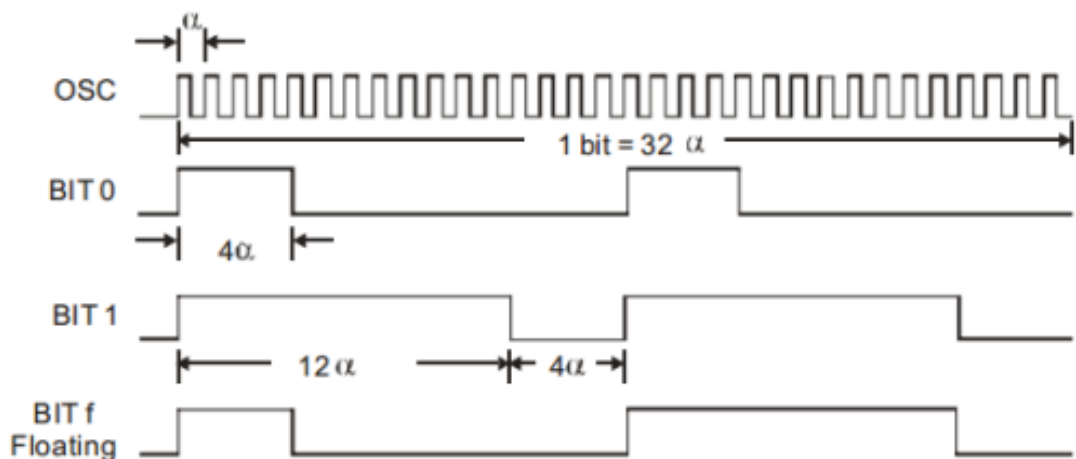
PT2262 adalah encoder remote control yang dipasangkan dengan teknologi CMOS PT2272. Ini mengkodekan data dan pin alamat menjadi bentuk kode serial yang sesuai untuk modulasi RF atau IR. PT 2262 memiliki maksimum 12 bit pin alamat tri-state yang menyediakan hingga 531.441 kode alamat; Sehingga secara drastis mengurangi tabrakan kode dan kemungkinan pemindaian kode yang tidak sah.

PT2262 mengkodekan alamat kode dan kumpulan data pada A0 ~ A5 dan A6 / D5 ~ A11 / D0 menjadi bentuk gelombang dan keluaran khusus adalah ke DOUT saat TE ditarik ke "0" (negara rendah). Bentuk gelombang ini diumpankan ke modulator RF atau pemancar IR untuk transmisi. Frekuensi radio atau sinar inframerah yang ditransmisikan diterima oleh demodulator RF atau penerima IR dan dibentuk kembali ke bentuk gelombang khusus. PT 2272 kemudian digunakan untuk memecahkan kode bentuk gelombang dan mengatur pin output yang sesuai. Jadi menyelesaikan pengkodean dan fungsi penentuan remote control.

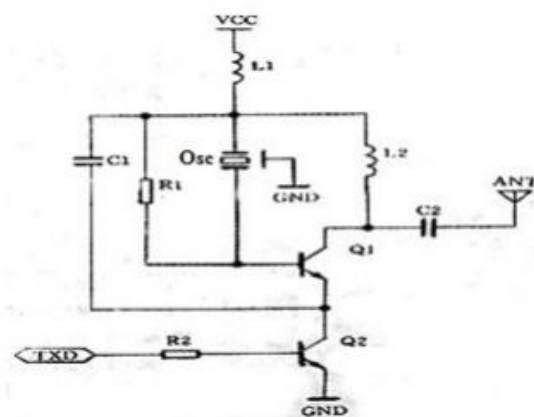
Sirkuit osilator build-in dari PT 2262 memungkinkan osilator presisi untuk dibangun dengan menghubungkan resistor eksternal antara pin OSC1 dan OSC2.

Bagi PT 2272 untuk memecahkan kode dengan benar bentuk gelombang yang diterima, frekuensi osilator PT 2272 harus 2,5 ~ 8 kali dari transmisi PT 2262.

Bit alamat / data dapat ditunjuk sebagai bit "0", "1", atau "f" jika berada dalam keadaan rendah, tinggi atau mengambang. Bentuk gelombang satu bit terdiri dari 2 siklus pulsa yang memiliki 16 periode waktu osilasi. Untuk keterangan lebih lanjut, lihat diagram di bawah ini:



Pasangan / komplemen dari IC PT2262 adalah IC PT2272 *remote control decoder* yang berfungsi sebagai pengurai sandi (*decoder*) signal yang disandikan oleh IC PT2262. Adapun skema *transmitter* RF PT2262 dapat dilihat pada gambar dibawah :



Konfigurasi dan keterangan konfigurasi IC PT2262 dapat dilihat pada Tabel di bawah ini :

Tabel 2.1 konfigurasi RF PT2262

Nama Pin	I/O	Deskripsi	Pin#
A0..A5	I	Pin masukan kode alamat A0..A5 Setiap pin dapat diset ke "0", "1", atau "F" (mengambang, <i>floating</i>)	1 – 6
A6/D5 ~ A11/D0	I	Pin masukan kode alamat A6..A11 atau pin data #5..#0 Saat digunakan untuk kode alamat, setiap pin dapat diset ke "0", "1", atau "F" (mengambang, <i>floating</i>). Apabila digunakan untuk data, pin ini hanya bisa diset ke "0" atau "1".	7 - 8 dan 10 – 13
OSC1	O	Pin <i>Oscillator</i> no. 1	15
OSC2	I	Pin <i>Oscillator</i> no. 2	16
DOUT	O	Pin keluaran (<i>Data Output Pin</i>) Setiap pin dapat diset ke "0", "1", atau "F" (mengambang, <i>floating</i>)	17
V _{cc}	-	Pin sumber daya, hubungkan dengan <i>positive power supply</i> (4 hingga 15 Volt)	18
V _{ss}	-	Hubungkan pin ini dengan <i>ground</i> (GND)	9

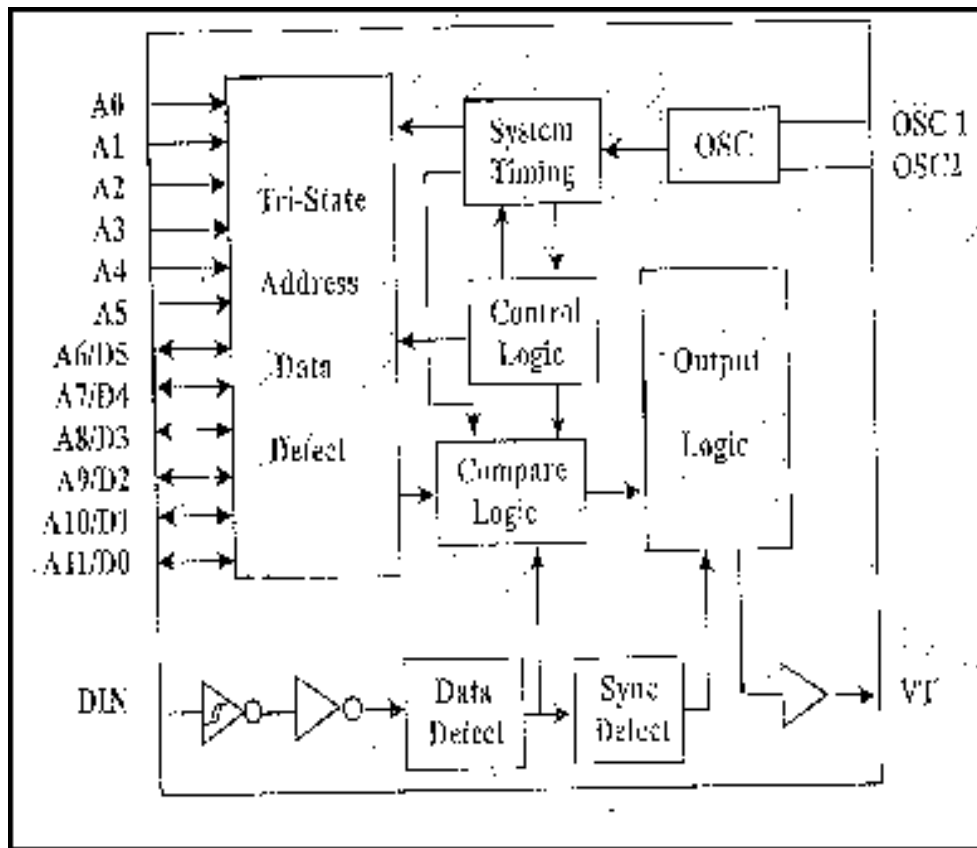
Karakteristik IC PT2262 sebagai berikut :

- Teknologi CMOS
- Konsumsi daya rendah
- Sampai 12 kode alamat Pin
- 6 data Pins
- V_{cc} = 4 ~ 15 Volts
- Tunggal resistor *Oscillator*

Bagian Pelaporan PT 2272

PT 2272 adalah remote control decoder yang dipasangkan dengan PT 2262 dengan menggunakan teknologi CMOS. Ini memiliki maksimum 12 bit pin alamat tri-state yang menyediakan hingga 531.441 kode alamat; Sehingga secara drastis mengurangi tabrakan kode dan kemungkinan pemindaian kode yang tidak sah. PT 2272 tersedia dalam beberapa pilihan yang sesuai dengan kebutuhan setiap aplikasi: jumlah pin output data, kait atau keluaran sesaat.

PT 2272 menafsirkan bentuk gelombang yang diterima dan dimasukkan ke dalam pin DIN. Bentuk gelombang diterjemahkan ke dalam kode kata yang berisi alamat, data dan bit sinkronisasi. Bit alamat yang didekode dibandingkan dengan alamat yang ditetapkan pada pin input alamat. Jika kedua alamat cocok untuk 2 kata kode berturut-turut, drive PT 2272:

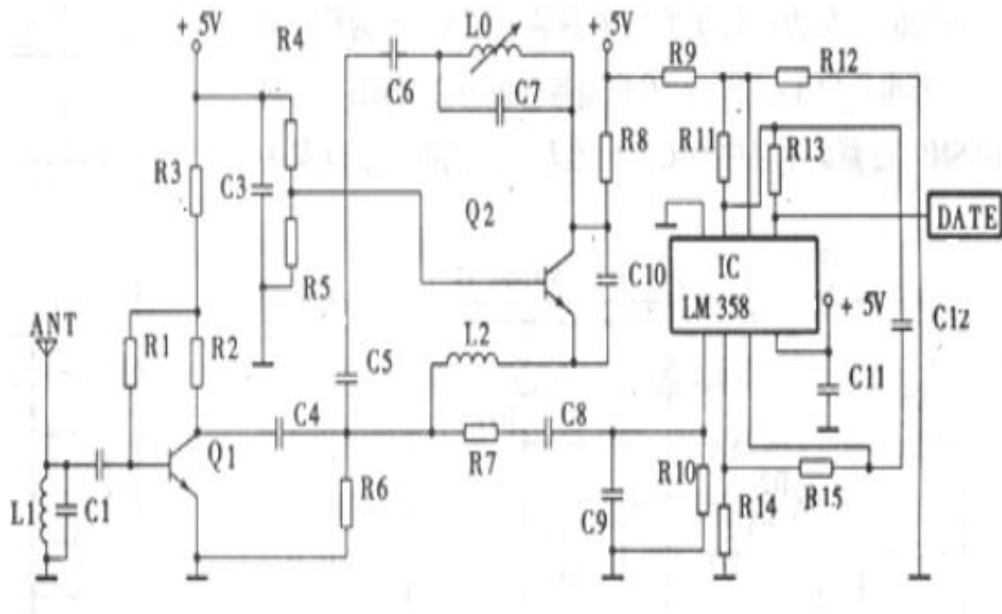


Pin keluaran data (s) yang bit data yang sesuai kemudian diterjemahkan menjadi bit "1", dan

Output VT ke tegangan tinggi (high state).

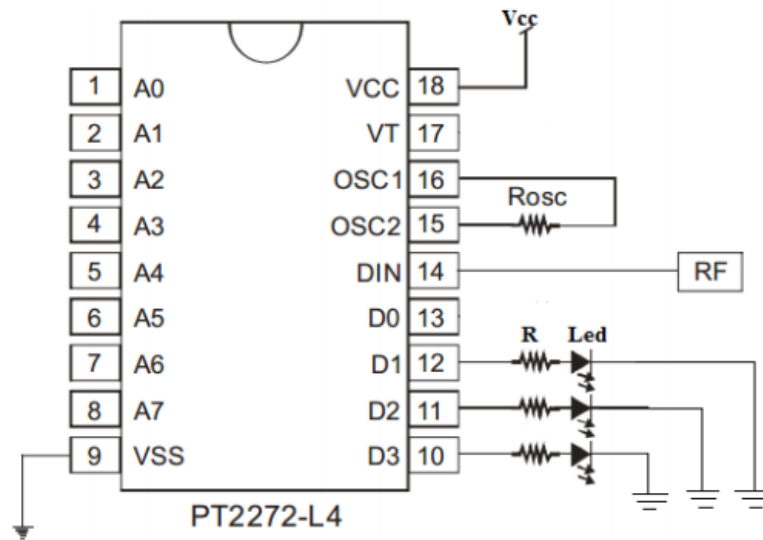
Sirkuit osilator build-in dari PT 22272 memungkinkan osilator presisi untuk dibangun hanya dengan resistor eksternal. Bagi PT 2272 untuk memecahkan kode dengan benar bentuk gelombang yang diterima, frekuensi osilator PT 2272 harus 2,5 ~ 8 kali lipat dari transmisi PT 2262.

Bit alamat / data dapat dirancang sebagai bit "0", "1" atau "f" jika berada dalam keadaan rendah, tinggi atau mengambang. Bentuk gelombang satu bit terdiri dari 2 siklus pulsa. Setiap pulsa memiliki 16 periode waktu osilasi. Untuk keterangan lebih lanjut, lihat diagram di bawah ini:



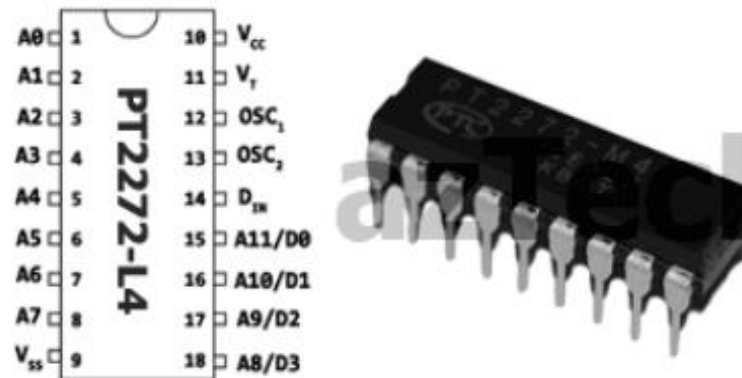
Gambar 2.10 Skema *Receiver* pada Sensor RF PT2272

Adapun rangkaian IC PT2272 dapat pada gambar 2.11 sebagai berikut :



Gambar 2.11 Rangkaian IC PT2272

Konfigurasi dan keterangan konfigurasi IC PT2272 dapat dilihat pada gambar 2.12 dan tabel 2.2 sebagai berikut :



Gambar 2.12 Konfigurasi IC PT2272

Tabel 2.2 Konfigurasi RF PT2272

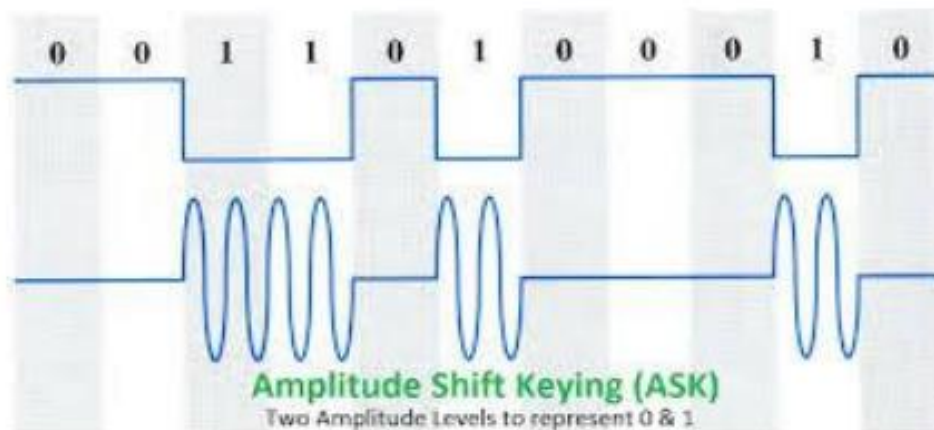
Pin Name	I/O	Description	Pin No.	
			18pins	20pins
A0 ~ A5	I	Code Address Pin Nos. 0 ~ 5. These six tri-state pins are detected by PT2272 to determine the encoded waveform bit 0 ~ bit 5. Each pin can be set to "0", "1", or "f" (floating).	1 ~ 6	1 ~ 6
A6/D5 ~ A11/D0	I/O	Code Address Pin Nos. 6 ~ 11/Data Pin Nos. 5 ~ 0. These six pins are used as higher address input bits or data output pins depending on the version (type) of PT2272 used. When used as address inputs, these pins are tri-state input pins and each pin can be set to "0", "1", or "f" (floating). When used as output pins, these pins are driven to VCC if (1) the address decoded from the waveform that was received matches the address setting at the address input pins, and (2) the corresponding data bits received is a "1" bit. Otherwise, they are driven to VSS.	7 ~ 8 10 ~ 13	7 ~ 8 12 ~ 15
DIN	I	Data Input Pin. The encoded waveform received is serially fed to PT2272 at this pin.	14	16
OSC 1	I	Oscillator Pin No. 1	15	17
OSC 2	O	Oscillator Pin No. 2	16	18
VT	O	Valid Transmission. Active High Signal. VT in high state signifies that PT2272 receives valid transmission waveform.	17	19
VCC	-	Positive Power Supply	18	20
VSS	-	Negative Power Supply	9	9
NC	-	No Connection	-	10 ~ 11

Karakteristik IC PT2272 sebagai berikut :

- Konsumsi daya rendah
- Teknologi TTL
- Sampai 12 Tri-State Kode Alamat Pin
- 6 data Pin

- $V_{cc} = 3 - 5$ Volt
- Tunggal resistor *Oscillator*

Pada *receiver* RF menggunakan sistem komunikasi *Amplitude Shift Keying* (ASK) yang berfungsi untuk menerima sinyal yang dikirimkan oleh *transmitter*. Sinyal yang dikirimkan berbentuk digital / kotak kemudian sinyal tersebut diubah kembali menjadi sinusoidal. Sinyal ASK yang terbentuk oleh *receiver* dapat dilihat pada gambar dibawah :



- **Gambar 2.13** Sinyal ASK pada *Receiver* IC PT2272

DataSheet Arduino Mega 2560

Arduino Mega 2560 adalah papan mikrokontroler berdasarkan ATmega2560 (datasheet). Ini memiliki 54 pin input / output digital (dimana 14 dapat digunakan sebagai output PWM), 16 input analog, 4 UART (port serial perangkat keras), osilator kristal 16 MHz, koneksi USB, colokan listrik, header ICSP, Dan tombol reset. Ini berisi semua yang dibutuhkan untuk mendukung mikrokontroler; Cukup hubungkan ke komputer dengan kabel USB atau nyalakan dengan adaptor AC-ke-DC atau baterai untuk memulai. Mega kompatibel dengan kebanyakan perisai yang dirancang untuk Arduino Duemilanove atau Diecimila.

Skema & Desain Referensi

File EAGLE: [arduino-mega2560-reference-design.zip](#)

Skema: [arduino-mega2560-skematis.pdf](#)

Ringkasan

Mikrokontroler	ATmega2560
Tegangan Operasi	5V
Tegangan Masukan (disarankan)	7-12V
Tegangan (batas)	6-20V
Pin I / O Digital	54 (yang 14 memberikan keluaran PWM)
Pin input analog	16
Arus DC per I / O Pin	40 mA
Arus DC untuk Pin 3.3V	50 mA
Flash Memory	256 KB dimana 8 KB digunakan oleh bootloader
SRAM 8 KB EEPROM	4 KB
Kecepatan Jam	16 MHz

Kekuasaan

Arduino Mega dapat bertenaga melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis.

Daya eksternal (non-USB) bisa datang baik dari adaptor AC-ke-DC (kutil dinding) atau baterai. Adaptor dapat dihubungkan dengan memasang konektor center-positive 2.1mm ke soket daya board. Memimpin dari baterai dapat dimasukkan ke dalam header pin Gnd dan Vin pada konektor POWER.

Papan dapat beroperasi pada suplai eksternal 6 sampai 20 volt. Jika dipasok dengan kurang dari 7V, pin 5V dapat memasok kurang dari lima volt dan board mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan mungkin terlalu panas dan merusak board. Kisaran yang disarankan adalah 7 sampai 12 volt.

Mega2560 berbeda dari semua papan sebelumnya karena tidak menggunakan chip driver USB-to-serial FTDI. Sebagai gantinya, fitur Atmega8U2 diprogram sebagai konverter USB-to-serial.

Pin daya adalah sebagai berikut:

- VIN. Tegangan masukan ke papan Arduino saat menggunakan sumber daya eksternal (berlawanan dengan 5 volt dari koneksi USB atau sumber listrik yang diatur lainnya). Anda bisa mensuplai voltase melalui pin ini, atau, jika mensuplai voltase melalui colokan listrik, aksesilah melalui pin ini.
- 5V. Catu daya yang diatur digunakan untuk menyalakan mikrokontroler dan komponen lainnya di papan tulis. Ini bisa datang baik dari VIN melalui regulator on-board, atau disediakan oleh USB atau suplai 5V yang diatur lainnya.
- 3V3 Pasokan 3,3 volt yang dihasilkan oleh regulator on-board. Maksimum saat ini adalah 50 mA.
- GND. Pin tanah

Memori

ATmega2560 memiliki memori flash 256 KB untuk menyimpan kode (dimana 8 KB digunakan untuk bootloader), 8 KB SRAM dan 4 KB dari EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Masukan dan keluaran

Masing-masing dari 54 pin digital pada Mega dapat digunakan sebagai input atau output, dengan menggunakan fungsi pinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada 5 volt. Setiap pin dapat menyediakan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal (terputus secara default) 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2: 17 (RX) dan 16 (TX); Serial 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pin 0 dan 1 juga dihubungkan ke pin pin ATmega8U2 USB-to-TTL Serial yang sesuai.
- Interupsi Eksternal: 2 (interupsi 0), 3 (interupsi 1), 18 (interupsi 5), 19 (interupsi 4), 20 (interupsi 3), dan 21 (interupsi 2). Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai rendah, tepi naik atau turun, atau perubahan nilai. Lihat fungsi AttachInterrupt () untuk rinciannya.

- PWM: 0 sampai 13. Sediakan output PWM 8 bit dengan fungsi `analogWrite ()`.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI dengan menggunakan perpustakaan SPI. Pin SPI juga terputus pada header ICSP, yang secara fisik kompatibel dengan Uno, Duemilanove dan Diecimila.
- LED: 13. Ada LED built-in yang terhubung ke pin digital 13. Bila pin bernilai HIGH, LED menyala, bila pinnya RENDAH, tidak menyala.
- I2C: 20 (SDA) dan 21 (SCL). Dukung komunikasi I2C (TWI) menggunakan perpustakaan `Wire` (dokumentasi di situs Wiring). Perhatikan bahwa pin ini tidak berada di lokasi yang sama dengan pin I2C pada Duemilanove atau Diecimila.

The Mega2560 memiliki 16 input analog, masing-masing memberikan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default mereka mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas jangkauan mereka menggunakan fungsi `AREF pin and analogReference ()`.

Ada beberapa pin lainnya di papan tulis:

- AREF. Tegangan referensi untuk input analog. Digunakan dengan `analogReference ()`.
- Reset. Bawah baris ini RENDAH untuk mereset mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset ke shields yang menghalangi yang ada di papan.

Komunikasi

The Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega2560 menyediakan empat perangkat keras UART untuk komunikasi serial TTL (5V). ATmega8U2 di papan menyalurkan salah satu dari USB ini dan menyediakan port com virtual untuk perangkat lunak di komputer (mesin Windows memerlukan berkas `.inf`, namun mesin OSX dan Linux akan mengenali papan sebagai port COM secara otomatis. Perangkat lunak Arduino Termasuk monitor serial yang memungkinkan data tekstual sederhana dikirim ke dan dari papan. LED RX dan TX di papan akan

berkedip saat data dikirimkan melalui chip ATmega8U2 dan koneksi USB ke komputer (tapi tidak untuk komunikasi serial Pin 0 dan 1).

Sebuah perpustakaan SoftwareSerial memungkinkan komunikasi serial pada salah satu pin digital Mega2560.

ATmega2560 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino mencakup perpustakaan Wire untuk menyederhanakan penggunaan bus I2C; Lihat dokumentasi di situs Wiring untuk rinciannya. Untuk komunikasi SPI, gunakanlah perpustakaan SPI.

Pemrograman

Mega Arduino dapat diprogram dengan perangkat lunak Arduino (download). Untuk detailnya, lihat referensi dan tutorialnya.

ATmega2560 di Arduino Mega hadir dengan bootloader yang memungkinkan Anda mengunggah kode baru ke dalamnya tanpa menggunakan programmer perangkat keras eksternal. Ini berkomunikasi menggunakan protokol STK500 yang asli (referensi, file header C).

Anda juga dapat melewati bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming); Lihat petunjuk ini untuk rinciannya.

Kode sumber firmware ATmega8U2 tersedia di gudang Arduino. ATmega8U2 dilengkapi dengan bootloader DFU, yang dapat diaktifkan dengan menghubungkan solder jumper di bagian belakang papan (dekat peta Italia) dan kemudian mengatur ulang 8U2. Anda kemudian dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau pemrogram DFU (Mac OS X dan Linux) untuk memuat firmware baru. Atau Anda bisa menggunakan header ISP dengan programmer eksternal (menimpa DFU bootloader). Lihat tutorial yang memberikan kontribusi pengguna ini untuk informasi lebih lanjut.

Reset Otomatis (Perangkat Lunak)

Melainkan meminta penekanan fisik tombol reset sebelum mengunggah, Arduino Mega2560 dirancang sedemikian rupa sehingga memungkinkannya diatur ulang oleh perangkat lunak yang berjalan pada komputer yang terhubung. Salah satu garis kontrol aliran perangkat keras (DTR) ATmega8U2 terhubung ke garis reset

ATmega2560 melalui kapasitor 100 nanofarad. Bila baris ini ditegaskan (diambil rendah), baris reset akan turun cukup lama untuk mengatur ulang chip. Perangkat lunak Arduino menggunakan kemampuan ini untuk memungkinkan Anda mengunggah kode dengan hanya menekan tombol upload di lingkungan Arduino. Ini berarti bahwa bootloader dapat memiliki batas waktu yang lebih pendek, karena penurunan DTR dapat dikoordinasikan dengan baik dengan dimulainya pengunggahan.

Penyiapan ini memiliki implikasi lain. Ketika Mega2560 terhubung ke komputer yang menjalankan Mac OS X atau Linux, ia me-reset setiap kali koneksi dibuat dari perangkat lunak (via USB). Untuk setengah detik berikutnya, bootloader berjalan di Mega2560. Meskipun diprogram untuk mengabaikan data yang salah (yaitu apa pun selain mengunggah kode baru), ini akan mencegat beberapa byte data pertama yang dikirim ke papan tulis setelah sambungan dibuka. Jika sketsa yang berjalan di papan menerima konfigurasi satu kali atau data lain saat pertama kali dimulai, pastikan perangkat lunak yang berkomunikasi menunggu satu detik setelah membuka sambungan dan sebelum mengirim data ini.

Mega2560 berisi jejak yang bisa dipotong untuk menonaktifkan auto-reset. Bantalan di kedua sisi jejak bisa disolder bersamaan untuk mengaktifkannya kembali. Ini berlabel "RESET-EN". Anda juga dapat menonaktifkan auto-reset dengan menghubungkan resistor 110 ohm dari 5V ke garis reset; Lihat thread forum ini untuk rinciannya.

Perlindungan Overcurrent USB

Arduino Mega2560 memiliki polibak yang dapat disetel ulang yang melindungi port USB komputer Anda dari celana pendek dan arus lebih. Meskipun kebanyakan komputer menyediakan perlindungan internal mereka sendiri, sekering menyediakan lapisan perlindungan ekstra. Jika lebih dari 500 mA diterapkan ke port USB, sekering akan secara otomatis memutus koneksi sampai pendek atau overload dilepaskan.

Karakteristik Fisik dan Kompatibilitas Perisai

Panjang maksimum dan lebar PCB Mega2560 masing-masing adalah 4 dan 2,1 inci, dengan konektor USB dan colokan listrik melampaui dimensi sebelumnya. Tiga lubang sekrup memungkinkan papan dipasang pada permukaan atau kotak. Perhatikan bahwa jarak antara pin 7 dan 8 digital adalah 160 mil (0,16"), bukan kelipatan dari jarak 100 mil dari pin lainnya.

Mega2560 dirancang agar kompatibel dengan kebanyakan perisai yang dirancang untuk Uno, Diecimila atau Duemilanove. Pin digital 0 sampai 13 (dan pin AREF dan GND yang berdekatan), input analog 0 sampai 5, header daya, dan header ICSP semuanya berada pada lokasi yang setara. Selanjutnya UART utama (port serial) terletak pada pin yang sama (0 dan 1), seperti halnya interupsi eksternal 0 dan 1 (pin 2 dan 3 masing-masing). SPI tersedia melalui header ICSP pada Mega2560 dan Duemilanove / Diecimila. Harap dicatat bahwa I2C tidak terletak pada pin yang sama pada Mega (20 dan 21) sebagai Duemilanove / Diecimila (input analog 4 dan 5).

Program keseluruhan

```
#include <Wire.h>           //library
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>

#define I2C_ADDR 0x3F // <<- Add your address here.
#define Rs_pin 0
#define Rw_pin 1
#define En_pin 2
#define BACKLIGHT_PIN 3 //register i2c LCD
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

#define off 0 //inisialisasi
#define on 1

#define buka 0
#define tutup 1
#define berhenti 2

#define Password_Lenght 7
```

```

char Data[Password_Lenght];
char Master[Password_Lenght] = "230817";
byte data_count = 0;
bool pass = false;
char customKey;

LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the symbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {22, 23, 24, 25}; //connect to the row pinouts of the
keypad
byte colPins[COLS] = {26, 27, 28, 29}; //connect to the column pinouts of the
keypad

//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);

int pin_remote = 8;
int pin_lock = 9;
int pin_motor1 = 10;
int pin_motor2 = 11;
int limit,limit_tutup,counter = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Wire.begin();

  pinMode(pin_remote,INPUT);
  pinMode(pin_lock,OUTPUT);
  pinMode(pin_motor1,OUTPUT);
  pinMode(pin_motor2,OUTPUT);
  pinMode(2,INPUT_PULLUP);
  pinMode(3,INPUT_PULLUP);
  pinMode(14,OUTPUT); //buzzer

```



```

lcd.begin (16,2); // <<-- our LCD is a 16x2, change for your LCD if needed
// LCD Backlight ON
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
lcd.setBacklight(HIGH);

lcd.home (); // go home on LCD
lcd.setCursor (0,0);
lcd.print("Gerbang_Otomatis");
lcd.setCursor (0,1);
lcd.print("Poltek Sriwijaya");

digitalWrite(pin_lock,LOW);
digitalWrite(pin_motor1,HIGH);
digitalWrite(pin_motor2,HIGH);
pass = false;
delay(1000);
lcd.clear();
}

void loop() {
// put your main code here, to run repeatedly:
int sensor = analogRead(15);
Serial.println(sensor);
keypad_input();
tombol();
if (pass == true){
lock(on);
lcd.clear();
do{
gate(buka);
limit = digitalRead(2);
lcd.setCursor(0, 0);
lcd.print("Password Benar");
lcd.setCursor(0,1);
lcd.print("Silahkan Masuk");
}
while (limit == HIGH);
gate(berhenti);
delay(1000);
pass = false;
lcd.clear();
}
else{
if(sensor > 300){ //aksi dari sensor getar

```

```

        counter= counter+1;
        if (counter >10){
            digitalWrite(14,HIGH);
            delay(1000);
            counter = 0;
        }
        Serial.println(counter);
    }
    pass = false;
    gate(berhenti);
    lock(off);
    digitalWrite(14,LOW);
}
Program Motor
void gate(int aksi)
{
    if(aksi == buka)
    {
        digitalWrite(pin_motor1,LOW);
        digitalWrite(pin_motor2,HIGH);
    }
    else if(aksi == tutup)
    {
        digitalWrite(pin_motor1,HIGH);
        digitalWrite(pin_motor2,LOW);
    }
    else{
        digitalWrite(pin_motor1,HIGH);
        digitalWrite(pin_motor2,HIGH);
    }
}

void lock(int aksi)
{
    if(aksi == on)
    {
        digitalWrite(pin_lock,LOW);
    }
    else{
        digitalWrite(pin_lock,HIGH);
    }
}

void tombol()
{

```

```

int key_open = digitalRead(pin_remote);
int key_close = digitalRead(7);
if(key_open == HIGH){
    pass = true;
    lcd.clear();
    lcd.setCursor (0,0);
    lcd.print("Gerbang_Otomatis");
    lcd.setCursor (0,1);
    lcd.print(" Kendali Remote ");
}
if(key_close == HIGH){
    lcd.clear();
    lcd.setCursor (0,0);
    lcd.print("Gerbang_Otomatis");
    lcd.setCursor (0,1);
    lcd.print(" Kendali Remote ");
    do{
        gate(tutup);
        limit_tutup = digitalRead(3);
    }
    while (limit_tutup == HIGH);
    lcd.clear();
}
}
}
Program keypad
void keypad_input()
{
    lcd.setCursor (0,0);
    lcd.print("=Enter Password=");
    customKey = customKeypad.getKey();
    if (customKey) // makes sure a key is actually pressed, equal to (customKey !=
NO_KEY)
    {
        Data[data_count] = customKey; // store char into data array
        lcd.setCursor(data_count,1); // move cursor to show each new char
        lcd.print("*"); // print char at said cursor
        data_count++; // increment data array by 1 to store new char, also keep track of
the number of chars entered
    }

    if(data_count == Password_Lenght-1) // if the array index is equal to the number
of expected chars, compare data to master
    {
        lcd.clear();
        lcd.setCursor(0, 0);

```

```

lcd.print("Password ");

if(!strcmp(Data, Master)) // equal to (strcmp(Data, Master) == 0)
{
  lcd.print("Benar");
  lcd.setCursor(0,1);
  lcd.print("Silahkan Masuk");
  pass = true;
}
else{
  lcd.print("Salah");
  lcd.setCursor(0,1);
  lcd.print("Coba Lagi");
  pass = false;
}
delay(1000);// added 1 second delay to make sure the password is completely
shown on screen before it gets cleared.
lcd.clear();
clearData();
}
}
void clearData()
{
  while(data_count !=0)
  { // This can be used for any array size,
    Data[data_count--] = 0; //clear array for new data
  }
  return;
}

```