

**RANCANG BANGUN OTOMATISASI PINTU GARASI DENGAN
KOMUNIKASI BLUETOOTH BERBASIS MIKROKONTROLER**



**Laporan ini Disusun Untuk Memenuhi Persyaratan
Menyelesaikan Pendidikan Diploma III Jurusan Teknik Komputer
Politeknik Negeri Sriwijaya Palembang**

Oleh:

Nama : Resti Febriani

Nim : 0612 3070 1288

**JURUSAN TEKNIK KOMPUTER
POLITEKNIK NEGERI SRIWIJAYA
PALEMBANG**

2015

**RANCANG BANGUN OTOMATISASI PINTU GARASI DENGAN
KOMUNIKASI BLUETOOTH BERBASIS MIKROKONTROLER**



Oleh:

Nama : Resti Febriani

Nim : 0612 3070 1288

Palembang, 2015

Disetujui Oleh,

Pembimbing I

Pembimbing II

Yulian Mirza, S.T., M.Kom

NIP. 196607121990031003

Ema Laila, S.Kom., M.Kom

NIP. 1977032920011220002

Mengetahui,

Ketua Jurusan Teknik Komputer

Ahyar Supani, S.T, M.T

NIP. 196802111992031002

ABSTRAK

RANCANG BANGUN OTOMATISASI PINTU GARASI DENGAN KOMUNIKASI BLUETOOTH BERBASIS MIKROKONTROLER

(Resti Febriani : 2015 : 41 Halaman)

Rancang bangun otomatisasi pintu garasi dengan komunikasi bluetooth berbasis mikrokontroler berfungsi untuk meningkatkan kenyamanan penghuni rumah dan dapat mengontrol pintu dari jarak jauh.. Dengan berkembangnya teknologi, telepon seluler bukan hanya sebatas digunakan untuk sms atau telepon saja, tapi dapat digunakan untuk mengendalikan sebuah sistem. Sistem kendali diimplementasikan untuk memberikan alternatif lain bagi manusia untuk menggerakkan atau mengendalikan suatu sistem dari jarak jauh. Rancang bangun pintu garasi dapat dikendalikan dengan menggunakan *handphone android* dengan aplikasi yang telah dibangun didalamnya. Pada sisi alat, terdapat sebuah perangkat *bluetooth* yang berfungsi untuk merespon data yang dikirimkan oleh *handphone android*. Mikrokontroler ATMega8535 yang digunakan sebagai unit pengolah data akan melakukan proses pengolahan data untuk menjalankan logika pengendalian.

Kata Kunci : Pintu Garasi, Bluetooth, Android, ATMega8535

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN PENGUJI	iii
MOTTO	iv
ABSTRAK	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
BAB II TINJAUAN UMUM	
2.1 Mikrokontroler	3
2.1.1 Sistem Mikrokontroler	3
2.2 Mikrokontroler ATmega 8535.....	3
2.2.1 Diagram Blok ATmega 8535	4
2.2.2 Fitur ATmega 8535	5
2.2.3 Konfigurasi pin ATmega 8535	5
2.3 Bluetooth.....	6
2.3.1 <i>Bluetooth Protocol Stack</i>	7
2.3.2 Topologi Jaringan Bluetooth.....	9
2.4 Android	9
2.5 Doorlock.....	11
2.6 Motor Servo	11
2.7 Relay	12
2.8 LED.....	12
2.9 Bahasa Pemrograman C	12
2.9.1 Tipe Data.....	13
2.9.2 Konstanta	14
2.9.3 Variabel	14
2.9.4 Deklarasi Variabel.....	15
2.9.5 Deklarasi Konstanta	15
2.9.6 Deklarasi Fungsi	15
2.9.7 Operasi Aritmatika	15
2.9.8 Operasi Logika	16
2.9.9 Komentar Program.....	16
2.10 Flowchart	17
2.11 <i>CodeVisionAVR C Compiler v1.24.7e</i>	18

BAB III RANCANG BANGUN

3.1	Tujuan Perancangan	23
3.2	Langkah-langkah Perancangan	24
3.3	Perancangan Blok Diagram.....	24
3.4	Rangkaian Pusat Pengendali	25
3.4.1	Perancangan Pengujian Rangkaian Bluetooth	26
3.4.2	Perancangan Pengujian Rangkaian Relay	26
3.4.3	Perancangan Pengujian Rangkaian Motor Servo.....	27
3.4.4	Perancangan Pengujian Rangkaian LED	27
3.4.5	Perancangan Pengujian Rangkaian Alat Keseluruhan ..	28
3.5	Pemilihan Komponen.....	28
3.6	Tahap Perancangan	30
3.6.1	Perancangan PCB.....	30
3.6.2	Perancangan Mekanik	31
3.7	Perancangan Software	31
3.8	Pembuatan Flowchart.....	31

BAB IV HASIL DAN PEMBAHASAN

4.1	Tahapan Pengujian Alat	34
4.2	Pengukuran Alat.....	38
4.2.1	Tujuan Pengukuran Alat	38
4.2.2	Peralatan yang Digunakan.....	38
4.2.3	Langkah-langkah Pengukuran.....	38
4.3	Hasil Pengukuran	39
4.3.1	Pengukuran Rangkaian Regulator LM7805.....	39
4.3.2	Pengukuran Tegangan Servo.....	39
4.4	Pembahasan	40

BAB V KESIMPULAN DAN SARAN

5.1	Kesimpulan	41
5.2	Saran.....	41

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1	Diagram Blok ATmega8535	4
Gambar 2.2	Konfigurasi Pin ATmega8535	5
Gambar 2.3	<i>Bluetooth Protocol Stack</i>	8
Gambar 2.4	Simbol Android	10
Gambar 2.5	Doorlock	11
Gambar 2.6	Motor Servo	11
Gambar 2.7	Relay	12
Gambar 2.8	LED	12
Gambar 2.9	Tampilan depan <i>CodeVisionAVR</i>	18
Gambar 3.1	Blok Diagram	24
Gambar 3.2	Rangkaian Sistem Minimum ATmega 8535	26
Gambar 3.3	Pengujian Rangkaian Bluetooth	26
Gambar 3.4	Pengujian Rangkaian Relay	27
Gambar 3.5	Pengujian Rangkaian Motor Servo	27
Gambar 3.6	Pengujian Rangkaian LED	28
Gambar 3.7	Rancangan Keseluruhan Rancang Bangun Otomatisasi Pintu Garasi dengan Komunikasi Bluetooth Berbasis Mikrokontroler	28
Gambar 3.8	Rancangan Design Alat Tampak Atas	31
Gambar 3.9	Flowchart Bluetooth Master	32
Gambar 3.10	Flowchart Rancang Bangun Otomatisasi Pintu Garasi dengan Komunikasi Bluetooth Berbasis Mikrokontroler	33
Gambar 4.1	Tampilan <i>file.apk</i> pada Android	34
Gambar 4.2	Tampilan menginstal <i>File.apk</i>	35
Gambar 4.3	Aplikasi Berhasil Dipasang	35
Gambar 4.4	Tampilan proses Membuka Aplikasi	36
Gambar 4.5	Tampilan Aplikasi Android	36
Gambar 4.6	Titik Ukur Rangkaian Regulator	39

DAFTAR TABEL

Tabel 2.1	<i>Bluetooth Protocol Stack</i>	7
Tabel 2.2	Tipe-tipe data Bahasa Pemrograman C.....	13
Tabel 2.3	Simbol-simbol Flowchart.....	17
Tabel 3.1	Daftar Komponen.....	29
Tabel 3.2	Daftar Alat dan Bahan.....	29
Tabel 4.1	Tabel Pengukuran Jarak Konektifitas Bluetooth.....	37
Tabel 4.2	Sistem Kerja Alat	37
Tabel 4.3	Tabel Pengukuran Regulator.....	39
Tabel 4.4	Hasil Pengukuran Servo	39

LISTING PROGRAM

/*

*/

This program was produced by the
CodeWizardAVR V2.05.0 Professional
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :
Version :
Date : 6/14/2015
Author : Resti Febriani
Company :
Comments:

Chip type : ATmega8535
Program type : Application
AVR Core Clock frequency: 11.000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256

*/

```
#include <mega8535.h>
```



```
#include <delay.h>
```

```
#ifndef RXB8
```

```
#define RXB8 1
```

```
#endif
```

```
#ifndef TXB8
```

```
#define TXB8 0
```

```
#endif
```

```
#ifndef UPE
```

```
#define UPE 2
```

```
#endif
```

```
#ifndef DOR
```

```
#define DOR 3
```

```
#endif
```

```
#ifndef FE
```

```
#define FE 4
```

```
#endif
```

```
#ifndef UDRE
```

```
#define UDRE 5
```

```
#endif
```

```
#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

char tampung='';

void ambil_bt();
// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif
```

```

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;

status=UCSRA;

data=UDR;

if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
// special case for receiver buffer size=256
if (++rx_counter == 0)
{
#else
if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;

rx_buffer_overflow=1;
}
#endif
}
}

```

```

    }

    tampung=data;
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;

while (rx_counter==0);

data=rx_buffer[rx_rd_index++];

#if RX_BUFFER_SIZE != 256
if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#endif

asm("cli")

--rx_counter;

asm("sei")

return data;
}

#pragma used-
#endif

// Standard Input/Output functions

```

```
#include <stdio.h>

// Declare your global variables here
unsigned char dat1,dat2,dat3;

void cek_lagi();
void ambil_bt();
void buka();
void tutup();
void servo();
void servo2();

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T
State2=T State1=T State0=T

PORTA=0x00;

DDRA=0xff;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T
State2=T State1=T State0=T

PORTB=0x00;

DDRB=0x00;

// Port C initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out
Func2=Out Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0
State2=0 State1=0 State0=0

PORTC=0x00;

DDRC=0xFF;

// Port D initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=Out Func0=Out

// State7=T State6=T State5=T State4=T State3=T
State2=T State1=0 State0=0

PORTD=0x00;

DDRD=0x03;

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=0x00;
```

```
TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off

TCCR1A=0x00;

TCCR1B=0x00;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
```



```
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1:
Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
```

```
TWCR=0x00;

// Global enable interrupts
#asm("sei")

while (1)
    {
        ambil_bt();
    }
}

// ambil perintah bt
void ambil_bt()
{
    while(1)
    {
        if (tampung=='A')
            {
                // PORTC.0=1;
                // PORTA.0=1;
                // PORTA.1=1;
                // PORTA.2=1;
                // delay_ms(1000);
                servo();
                break;
            }
    }
}
```

```
if (tampung=='B')
{
    //PORTC.0=0;
    //PORTA.0=0;
    //PORTA.1=0;
    //PORTA.2=0;
    //putchar (20);
    //delay_ms(1000);
    servo2();
    break;
}
// PORTC.0=0;
// PORTA.0=0;
//PORTA.1=0;
//PORTA.2=0;
putchar (10);
delay_ms(1000);
break;

}

}

//servo
void servo()
{
    while(1)
```

```
{  
    PORTA.0=1;  
    PORTA.1=1;  
    delay_ms(3000);  
    tutup();  
    PORTA.0=0;  
    PORTA.1=0;  
    delay_ms(3000);  
    break;  
}  
  
}  
  
void servo2()  
{  
    while(1)  
    {  
  
        PORTA.0=1;  
        PORTA.1=1;  
        delay_ms(3000);  
        buka();  
        PORTA.0=0;  
        PORTA.1=0;  
        delay_ms(3000);  
        break;  
    }  
}
```

```

}

void tutup()
{
    while(1)
    {
        int i;
        for (i=0; i<290; i++)
        {
            PORTC.0=1;          //memberikan pulsa high
            delay_ms(0.9);      //pulsa high diberikan
selama 0.9 ms
            PORTC.0=0;
            delay_ms(20);
        }
        break;
    }
}

void buka()
{
    while(1)
    {
        int i;
        for (i=0; i<280; i++)
        {

```

```
        PORTC.0=1;           //memberikan pulsa high
        delay_ms(2.1);      //pulsa high diberikan
selama 2.1 ms
        PORTC.0=0;
        delay_ms(20);
    }
    break;
}
}
```