LAMPIRAN

A. Skrip Program Akses Sensor Ultrasonik *Slave Microcontroller* Atmega 328P

```
#include <mega328p.h>
#include <stdio.h>
#include <delay.h>

//===Define Makro Pin Ping=====
#define trigger0    PORTB.2
#define echo0       PINB.2
#define direction0  DDRB.2
#define trigger1    PORTB.1
#define echo1       PINB.1
#define direction1  DDRB.1
#define trigger2    PORTB.0
#define echo2       PINB.0
#define direction2  DDRB.0
#define trigger3    PORTD.7
#define echo3       PIND.7
#define direction3  DDRD.7
#define trigger4    PORTD.2
#define echo4       PIND.2
#define direction4  DDRD.2
#define trigger5    PORTD.3
#define echo5       PIND.3
#define direction5  DDRD.3
#define batasPulse  200
#define delayPulse 5


void ping0(){
   unsigned int pulse= 0,data= 0;
   pulse= 0; direction0= 1; trigger0= 1;delay_us(5); trigger0= 0;direction0= 0;
   while(echo0== 0);
   while(echo0== 1){
      pulse++;
      delay_us(58);
      if(pulse>=batasPulse){
         while(echo0==1);
         break;
      }  }
   data=pulse;putchar(data);delay_ms(delayPulse);}
```

```
void ping1(){
   unsigned int pulse= 0, data= 0;
   pulse= 0;
   direction1= 1;  trigger1= 1; delay_us(5); trigger1= 0; direction1= 0;
   while(echo1== 0);
   while(echo1== 1){
      pulse++;  delay_us(58);
      if(pulse>=batasPulse){
         while(echo1== 1);break;}}
   data= pulse; putchar(data); delay_ms(delayPulse);}

void ping2(){
  unsigned int pulse= 0, data= 0;
  pulse= 0; direction2= 1;trigger2= 1; delay_us(5); trigger2= 0;direction2= 0;
   while(echo2== 0){};
   while(echo2== 1){
   pulse++;
      delay_us(58);
      if(pulse>=batasPulse){
         while(echo2== 1); break;
      } }
   data= pulse; putchar(data); delay_ms(delayPulse);}

void ping3(){
   unsigned int pulse= 0, data= 0;
   pulse= 0; direction3= 1;trigger3= 1; delay_us(5); trigger3= 0; direction3= 0;
   while(echo3== 0){};
   while(echo3== 1){
      pulse++;
         while(echo3==1);
         break;
      }}
   data= pulse;
   putchar(data);
   delay_ms(delayPulse);}

void ping4(){
   unsigned int pulse= 0, data= 0;
   pulse= 0; direction4= 1; trigger4= 1; delay_us(5);  trigger4= 0;   direction4= 0;
   while(echo4== 0){};
   while(echo4== 1){
      delay_us(58);
      if(pulse>=batasPulse){
      pulse++;
      delay_us(58);
      if(pulse>=batasPulse){
         while(echo4==1);
         break;}  }
   data= pulse;
   putchar(data);
   delay_ms(delayPulse);
}

void ping5(){
   unsigned int pulse= 0, data= 0;
   pulse= 0; direction5= 1; trigger5= 1;  delay_us(5);trigger5= 0;  direction5= 0;
   while(echo5== 0){};
   while(echo5== 1){
      pulse++;
      delay_us(58);
      if(pulse>=batasPulse){
         while(echo5==1);
         break;} }
   data= pulse;
   putchar(data);
   delay_ms(delayPulse);}

void main(void) {
//===USART TX INIT Mikro slave Ping Ultrasonik
UCSR0A=(0<<RXC0) | (0<<TXC0) | (0<<UDRE0) | (0<<FE0) | (0<<DOR0) | (0<<UPE0) |
(0<<U2X0) | (0<<MPCM0);
UCSR0B=(0<<RXCIE0) | (0<<TXCIE0) | (0<<UDRIE0) | (0<<RXEN0) | (1<<TXEN0) |
(0<<UCSZ02) | (0<<RXB80) | (0<<TXB80);
UCSR0C=(0<<UMSEL01) | (0<<UMSEL00) | (0<<UPM01) | (0<<UPM00) | (0<<USBS0) |
(1<<UCSZ01) | (1<<UCSZ00) | (0<<UCPOL0);
UBRR0H=0x00;
UBRR0L=0x07;
while (1) {
      putchar(0);ping0(); ping1();  ping2();  ping3(); ping4();   ping5();
     }
}
```

B. Skrip Program Utama *Master Microcontroller* Atmega 2560

1. Skrip Define Makro Program untuk Kalibrasi

```
#include <mega2560.h>
#include <stdio.h>
#include <delay.h>
#include <i2c.h>
#include <alcd.h>

//======makro Tombol Menu=============
#define    start          !PINC.6
#define    ok             !PINA.7
#define    scroll_up       !PINA.5
#define    scroll_down      !PINA.3
//Start robot dengan frek sound
#define    sound         ((data_count)>300&&(data_count)<400)

//====Makro Kalibrasi Batas Nilai Ambang ADC Pembacaan Lantai Arena=====
#define  deteksiPutih0      (s0>=185 && s1<=254)  //Segar depan baca warna putih
#define  deteksiPutih1      (s1>=185 && s1<=254)  //Segar tengah baca warna putih
#define  deteksiPutih2      (s2>=185 && s2<=254)  //Segar belakang baca warna putih
#define  deteksiGrey0       (grs0>=100 && grs0<=140)  //Segar depan baca warna grey
#define  deteksiGrey1       (grs1>=100 && grs1<=140)  //Segar tengah  baca warna grey
#define  deteksiGrey2       (grs2>=100 && grs2<=140)  //Segar belakang baca warna
grey
#define  deteksiHitam0      (grs0>=20 && grs0<=90)   //Segar depan baca warna hitam
#define  deteksiHitam1      (grs1>=20 && grs1<=90)  //Segar tengah  baca warna hitam
#define  deteksiHitam2      (grs2>=20 && grs2<=90)   //Segar belakang baca warna
hitam

//====Makro Kalibrasi Parameter Checkpoint Ganti Telusur Di lorong====
#define  arah1 ((ruang==1 && deteksiGrey0  && deteksiHitam1  &&  deteksiHitam2)
&& ((ping2>=130 && ping2<=140) && (ping5>=75 && ping5<=85)))
```

```
#define  arah2 ((ruang==1 && deteksiHitam0 && deteksiGrey1  && deteksiGrey2) &&
((ping2>=75  &&  ping2<=85) && (ping5>=130 && ping5<=140)))
//makro untuk kondisi ganti  posisi 1 atau posisi 2
#define  checkpoint _lorong   (arah1 | | arah2)
```

2. Skrip Inisialisasi Variabel Program

```
unsigned int ping0, ping1, ping2, ping3, ping4, ping5, sharp0, s0, s1, s2;
unsigned char cr=0, statusCek_ruang=0,last_statusCek_ruang=0;
unsigned char cp=0, statuscp=0, last_statuscp=0;
int error;
unsigned char dataPutar;
eeprom unsigned int setpointKa=22, setpointKi=22;
eeprom unsigned int kp=55, ts=1, kd= 23;
float derror, l_derror, Vd, Vp, R_PID;
```

3. Program Inisialisasi *UART RX Interrupt*

```
//=======UART INTERRUPT INIT=========
#define DATA_REGISTER_EMPTY (1<<UDRE0)
#define RX_COMPLETE (1<<RXC0)
#define FRAMING_ERROR (1<<FE0)
#define PARITY_ERROR (1<<UPE0)
#define DATA_OVERRUN (1<<DOR0)

// USART1 Receiver buffer
#define RX_BUFFER_SIZE1 6
char rx_buffer1[RX_BUFFER_SIZE1];
#if RX_BUFFER_SIZE1 <= 256
unsigned char rx_wr_index1=0,rx_rd_index1=0;
#else unsigned int rx_wr_index1=0,rx_rd_index1=0;
#endif
#if RX_BUFFER_SIZE1 < 256
unsigned char rx_counter1=0;
#else

unsigned int rx_counter1=0;
#endif
```

```
bit rx_buffer_overflow1;

interrupt [USART1_RXC] void usart1_rx_isr(void){
char status,data;
status=UCSR1A;
data=UDR1;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
  {
  if(data==0) rx_wr_index1=0;
  else{
  rx_buffer1[rx_wr_index1++]=data;
  ping0= rx_buffer1[0];ping1= rx_buffer1[1];ping2= rx_buffer1[2];ping3= rx_buffer1[3];
  ping4= rx_buffer1[4]; ping5= rx_buffer1[5];
  if (rx_wr_index1 == RX_BUFFER_SIZE1) rx_wr_index1=0;}
  }
}
```

4. Program Akses *Register* ADC dan Akses Sensor Garis LED-Photodioda

```
// ADC Acess Register
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (1<<ADLAR))
unsigned int read_adcc(unsigned char channel)
{
ADMUX &= 0xE0;          //Clear bits MUX0-4
ADMUX |= channel&0x07;   //Defines the new ADC channel to be read by setting bits
MUX0-2
ADCSRB = channel&(1<<3); //Set MUX5
//delay_us(10);
ADCSRA |= (1<<ADSC);      //Starts a new conversion
while(ADCSRA & (1<<ADSC));  //Wait until the conversion is done
return ADCH;
}

void read_segar(){  s0=read_adcc(0);    s1=read_adcc(2); s2=read_adcc(4);}
```

5. Program Pencacah Waktu Menggunakan *Timer Interrupt*

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void){

TCNT0=0x70;
   if(fsound==1)  {
      cacah++;
         if (cacah==10)
         {  cacah=0; data_count=count;      count=0;} }
   if(counterpid==1) // timer 100 ms{
      cpid++;
      if(cpid==10) //untuk mengirim data setiap 100 ms (10 ms x 10) {
       printf("%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d \n",ccpid,
       cchange,grs0,grs1,grs2,ruang,cek_ruang,cp,ping2,ping5,ping1,ping3,sharp0,
       ping0,ping4);
         cpid=0;ccpid++;  }
   }
   if(counterchange==1)// timer 1 detik{
      ccacah++;
      if (ccacah==100)  {
            ccacah=0;change++;
         } }}
```

6. Program Akses *PORT I/O* utama , *Register Timer*, dan *Register UART*

```
void inisialisasi(){
//inisialisasi Pin Port I/O
   PORTA   =0b10001010;  DDRA    =0b01000000;
   PORTB   =0b00000000;  DDRB    =0b00000000;
   PORTC   =0b01100001;  DDRC    =0b10101010;
   PORTD   =0b00000000;  DDRD    =0b10000000;
   PORTE   =0b00000000;  DDRE     =0b00000000;
   PORTF   =0b00000000;   DDRF    =0b00000000;
   PORTG   =0b00000000;  DDRG    =0b00000000;
   PORTH   =0b00000000;  DDRH    =0b00000000;
   PORTI    =0b00000000;   DDRI     =0b00000000;
   PORTJ    =0b00000000;   DDRJ    =0b00000000;
   PORTK    =0b00000000;  DDRK   =0b00000000;
   PORTL    =0b00000000;  DDRL    =0b00000100;

// Timer/Counter 0 initialization
TCCR0A=(0<<COM0A1) | (0<<COM0A0) | (0<<COM0B1) | (0<<COM0B0) |
(0<<WGM01) | (0<<WGM00);
TCCR0B=(0<<WGM02) | (1<<CS02) | (0<<CS01) | (1<<CS00);
TCNT0=0x70; OCR0A=0x00; OCR0B=0x00;

// Timer/Counter 0 Interrupt(s) initialization
TIMSK0=(0<<OCIE0B) | (0<<OCIE0A) | (1<<TOIE0);

//USART 1 INIT
UCSR1A=(0<<RXC1) | (0<<TXC1) | (0<<UDRE1) | (0<<FE1) | (0<<DOR1) | (0<<UPE1) |
(0<<U2X1) | (0<<MPCM1);
UCSR1B=(1<<RXCIE1) | (0<<TXCIE1) | (0<<UDRIE1) | (1<<RXEN1) | (1<<TXEN1) |
(0<<UCSZ12) | (0<<RXB81) | (0<<TXB81);
UCSR1C=(0<<UMSEL11) | (0<<UMSEL10) | (0<<UPM11) | (0<<UPM10) | (0<<USBS1) |
(1<<UCSZ11) | (1<<UCSZ10) | (0<<UCPOL1);
UBRR1H=0x00; UBRR1L=0x07;

lcd_init(16);
#asm("sei")
}
```

7. Program *Mapping* Parameter Garis dan *Checkpoint* pada Labirin

```
void cacah_garisPutih(){
    read_segar();
    statusCek_ruang = deteksiPutih2;
    if (statusCek_ruang != last_statusCek_ruang) {
        if (statusCek_ruang == 1){cr++;}
        delay_ms(50); last_statusCek_ruang = statusCek_ruang;
        }
}

void cacah_checkpoint_maze_mapping(){
   statuscp = checkpoint_lorong;
    if (statuscp!= last_statuscp){
     if (statuscp == 1) {  cp++;}
     delay_ms(50); last_statuscp = statuscp;
     delay_ms(200);
     }
    if(cp>1){cp=0;}
}
```

8. Program Navigasi *Wall Following* dan Kontrol PID (*Proportional Derivative)*

```
void errorKa(){
   error=ping3-setpointKa;
   if(error  >  10)              {error= 24; goto bawah1;}
   if(error  >=  9 && error  <= 10) {error= error +10;goto bawah1;}
   if(error  >=  6 && error  <=  8) {error= error +3;goto bawah1;}
   if(error  >=  3 && error  <=  5) {error= error +2;goto bawah1;}
   if(error  >=  1 && error  <=  2) {error= error +2;goto bawah1;}
   if(error  ==  0)                         {error= 1;goto bawah1;}
   if(error  <= -1 && error  >= -2) {error= error -2;goto bawah1;}
   if(error  <= -3 && error  >= -5) {error= error -2;goto bawah1;}
   if(error  <= -6 && error  >= -8) {error= error -3;goto bawah1;}
   if(error  <= -9 && error  >=-10) {error= error -10;goto bawah1;}
   if(error  < -10)              {error= -24;goto bawah1;}
   bawah1:
   derror=(float)1/error;
   Vp = (float)kp*derror*10;  Vd = (float)(kd/ts)*(derror - l_derror);
   R_PID = (float)(Vp + Vd);
   l_derror=derror;
}
void errorKi(){
   error= setpointKi-ping1;
   if(error  >   10)             {error= 24; goto bawah2;}
   if(error  >=  9 && error  <= 10) {error= error +10;goto bawah2;}
   if(error  >=  6 && error  <=  8) {error= error +3;goto bawah2;}
   if(error  >=  3 && error  <=  5) {error= error +2;goto bawah2;}
   if(error  >=  1 && error  <=  2) {error= error +2;goto bawah2;}
   if(error  ==  0)                         {error= 1;goto bawah2;}
   if(error  <= -1 && error  >= -2) {error= error -2;goto bawah2;}
   if(error  <= -3 && error  >= -5) {error= error -2;goto bawah2;}
   if(error  <= -6 && error  >= -8) {error= error -3;goto bawah2;}
   if(error  <= -9 && error  >=-10) {error= error -10;goto bawah2;}
   if(error  < -10)              {error= -24; goto bawah2;}
   bawah2:
   derror=(float)1/error;
   Vp = (float) kp*derror*10; Vd = (float)(kd/ts)*(derror - l_derror);
   R_PID = (float)(Vp+Vd);
   l_derror=derror;
}
```

```
//============WALL FOLLOWER  NAVIGATION================
void telusur_kanan(char batas_kanan,char batas_depan)
{
   dataPutar= 10;  errorKa();  konversijarak();
   if(ping2<batas_kanan ||(sharp0<batas_depan && sharp0>0))  {
   dataPutar=25; }
   putchar1(R_PID);  putchar1(dataPutar); putchar1(0);dataPutar= 0;
}

void telusur_kiri(char batas_kiri,char batas_depan)
{
   dataPutar= 15;
   errorKi();   konversijarak();
   if(ping2<batas_kiri||(sharp0<batas_depan && sharp0>0))  {
   dataPutar=20;    }
   putchar1(R_PID); putchar1(dataPutar);  putchar1(0);dataPutar=0;
}
```

9. Program Perintah Manuver Dasar Robot Via UART ke *Slave Microcontroller* Atmega32

```
void muter_kanan()  {putchar1(255);putchar1(20);putchar1(0);}
void muter_kiri()       {putchar1(255);putchar1(25);putchar1(0);}
void muter_kanan_scan_api() {putchar1(255);putchar1(70);putchar1(0);}
void muter_kiri_scan_api()      {putchar1(255);putchar1(75);putchar1(0);}
void maju()               {putchar1(255);putchar1(35);putchar1(0);}
void mundur()           {putchar1(255);putchar1(30);putchar1(0);}
void geser_kanan()   {putchar1(255);putchar1(50);putchar1(0);}
void geser_kiri()        {putchar1(255);putchar1(55);putchar1(0);}
void reset_posisi()   {putchar1(255) ;putchar1(5);putchar1(0);}
```

10. Skrip Program Navigasi dengan Algoritma *Maze Mapping*

```
void maze_mapping_algorithm){
   while((ping2<25) || (sharp0<25 && sharp0>0)){
   cacah_garisPutih(); cacah_checkpoint_maze_mapping();
   konversijarak();    muter_kanan();
   }

   while(1){
   normal:
      cacah_garisPutih();
      cacah_checkpoint_maze_mapping();
      konversijarak();
        normal_telusur1:
        if(cp==0 && cr==0){
        telusur_kiri(18,18);
        }

         if(cp==0 && cr==2){
            telusur_kiri(18,18);
            if(cr==2){
               for(m=0;m<5;m++){
               telusur_kiri(18,18);delay_ms(100);}
               for(m=0;m<4;m++){
               reset_posisi();delay_ms(100);}
               for(m=0;m<38;m++){
               muter_kanan_scan_api();delay_ms(100);
               }
               if(ping0>ping4){
               for(m=0;m<8;m++){
               geser_kiri();delay_ms(100);}
               goto lanjut1;
               }
               lanjut1:
               cr=0;
               for(m=0;m<6;m++){
               reset_posisi();delay_ms(100);}
               goto normal_telusur1;
            }
         }
         if(cp==0 && cr==1){
            telusur_kiri(19,19);
         }

         if(cp==1 && cr==1){
            telusur_kanan(19,19);
         }

        normal_telusur2:
        if(cp==1 && cr==0){
        telusur_kanan(18,18);
        }

         if(cp==1 && cr==2){
            telusur_kanan(18,18);
            if(cr==2){
               for(m=0;m<5;m++){
               telusur_kanan(18,18);delay_ms(100);}
               for(m=0;m<4;m++){
               reset_posisi();delay_ms(100);}
               for(m=0;m<38;m++){
               muter_kiri_scan_api();delay_ms(100);}

               if(ping0<ping4){
               for(m=0;m<8;m++){
               geser_kanan();delay_ms(100);}
               goto lanjut2;
               }
               lanjut2:
               cr=0;
               for(m=0;m<6;m++){
               reset_posisi();delay_ms(100);}
               goto normal_telusur2;
            }
         }

      lcd_gotoxy(0,0); sprintf(lcd_buffer,"T:%d  CR:%d ",cchange,cr);
      lcd_puts(lcd_buffer); lcd_gotoxy(0,1);   sprintf(lcd_buffer,"Scp:%d cp:%d”,
      statuscp,cp);lcd_puts(lcd_buffer);
   }
}
```

11. Skrip Program Utama (*Main Program)*

```
void main(void){
inisialisasi();reset_posisi();mulai();
lcd_clear();lcd_gotoxy(0,0);lcd_putsf("===Bismillah===");delay_ms(400);lcd_gotoxy(0,0
);
lcd_putsf("=Go MR.COOL MK7=");delay_ms(300);counterpid=1;counterchange=1;
lcd_clear();
 while(1){
       while(pilihAlgoritma==0){ telusur_kanan ();}
       while(pilihAlgoritma==1){ telusur_kiri ();}
       while(pilihAlgoritma==2){maze_mapping_algorithm();}
   }
}
```