Lampiran 1 Datasheet Arduino Mega 2560

Arduino MEGA 2560

CE

Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), 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 Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

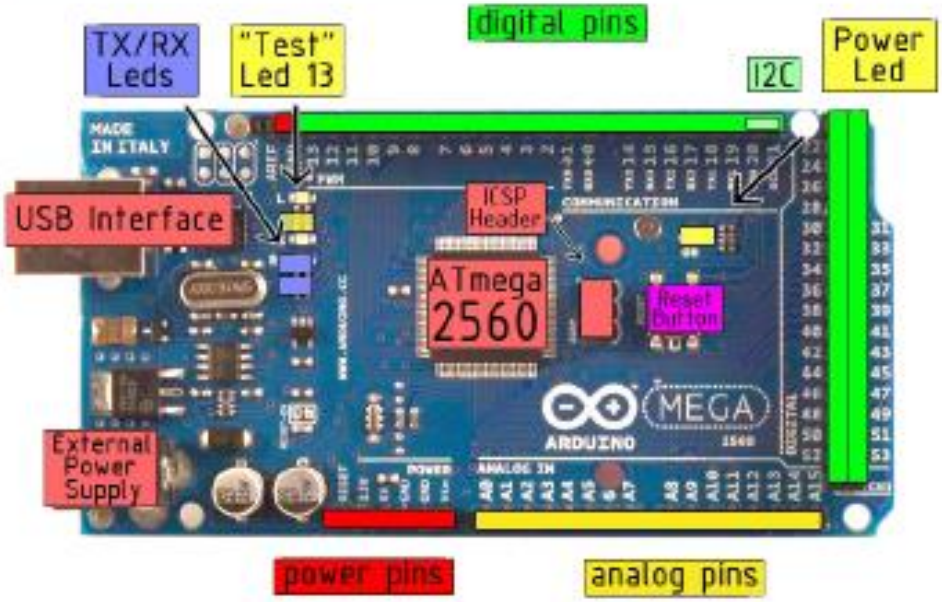
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Lampiran 2 Datasheet ACS 712

ACS712

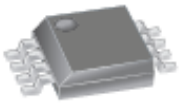
Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor

Features and Benefits

- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5 µs output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at T_A = 25°C
- Small footprint, low-profile SOIC8 package
- 1.2 mΩ internal conductor resistance
- 2.1 kVRMS minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratiometric output from supply voltage

TÜV America Certificate Number: U8V 06 05 54214 010

Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1

Description

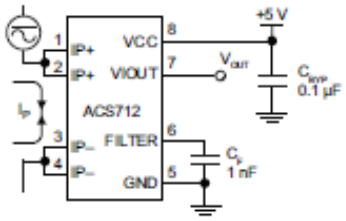
The Allegro™ ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switch-mode power supplies, and overcurrent fault protection. The device is not intended for automotive applications.

The device consists of a precise, low-offset, linear Hall circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which the Hall IC converts into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope (>$V_{IOUT(Q)}$) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sampling. The internal resistance of this conductive path is 1.2 mΩ typical, providing low power loss. The thickness of the copper conductor allows survival of

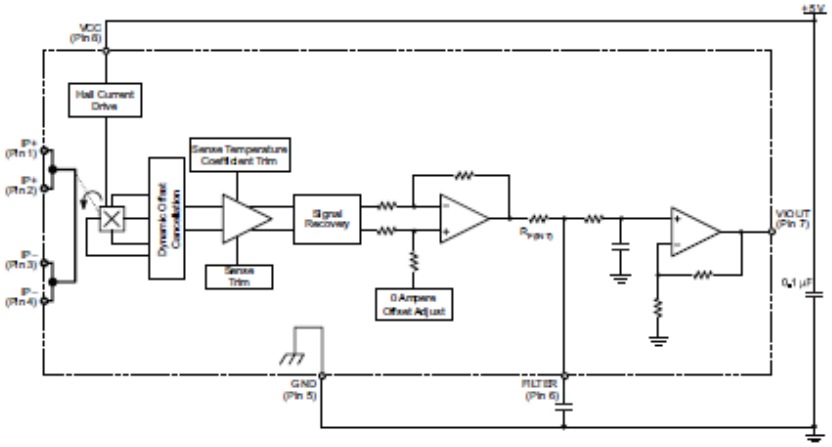
Continued on the next page...

Typical Application

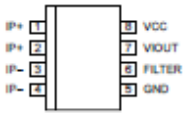


Application 1. The ACS712 outputs an analog signal, V_{OUT}, that varies linearly with the uni- or bi-directional AC or DC primary sampled current, I_P, within the range specified. C_F is recommended for noise management, with values that depend on the application.

Functional Block Diagram



Pin-out Diagram



Terminal List Table

Number	Name	Description
1 and 2	IP+	Terminals for current being sampled; fused internally
3 and 4	IP–	Terminals for current being sampled; fused internally
5	GND	Signal ground terminal
6	FILTER	Terminal for external capacitor that sets bandwidth
7	VIOUT	Analog output signal
8	VCC	Device power supply terminal

COMMON OPERATING CHARACTERISTICS[1] over full range of T_A, C_F = 1 nF, and V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
ELECTRICAL CHARACTERISTICS						
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Supply Current	I_{CC}	V_{CC} = 5.0 V, output open	–	10	13	mA
Output Capacitance Load	C_{LOAD}	VIOUT to GND	–	–	10	nF
Output Resistive Load	R_{LOAD}	VIOUT to GND	4.7	–	–	kΩ
Primary Conductor Resistance	$R_{PRIMARY}$	T_A = 25°C	–	1.2	–	mΩ
Rise Time	t_r	I_P = I_P(max), T_A = 25°C, C_{OUT} = open	–	3.5	–	µs
Frequency Bandwidth	f	–3 dB, T_A = 25°C; I_P is 10 A peak-to-peak	–	80	–	kHz
Nonlinearity	E_{LIN}	Over full range of I_P	–	1.5	–	%
Symmetry	E_{SYM}	Over full range of I_P	98	100	102	%
Zero Current Output Voltage	$V_{IOUT(Q)}$	Bidirectional; I_P = 0 A, T_A = 25°C	–	V_{CC} x 0.5	–	V
Power-On Time	t_{PO}	Output reaches 90% of steady-state level, T_J = 25°C, 20 A present on leadframe	–	35	–	µs
Magnetic Coupling[2]			–	12	–	G/A
Internal Filter Resistance[3]	$R_{F(INT)}$			1.7		kΩ

[1]Device may be operated at higher primary current levels, I_P, and ambient, T_A, and internal leadframe temperatures, T_A, provided that the Maximum Junction Temperature, T_J(max), is not exceeded.

[2]1G = 0.1 mT.

[3]$R_{F(INT)}$ forms an RC circuit via the FILTER pin.

COMMON THERMAL CHARACTERISTICS[1]

			Min.	Typ.	Max.	Units
Operating Internal Leadframe Temperature	T_A	E range	–40	–	85	°C
					Value	Units
Junction-to-Lead Thermal Resistance[2]	$R_{\theta JL}$	Mounted on the Allegro ASEK 712 evaluation board			5	°C/W
Junction-to-Ambient Thermal Resistance	$R_{\theta JA}$	Mounted on the Allegro 85-0322 evaluation board, includes the power consumed by the board			23	°C/W

[1]Additional thermal information is available on the Allegro website.

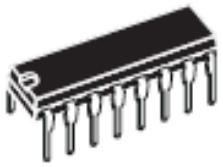
[2]The Allegro evaluation board has 1500 mm² of 2 oz. copper on each side, connected to pins 1 and 2, and to pins 3 and 4, with thermal vias connecting the layers. Performance values include the power consumed by the PCB. Further details on the board are available from the Frequently Asked Questions document on our website. Further information about board design and thermal performance also can be found in the Applications Information section of this datasheet.

Lampiran 3 Datasheet ULN 2003

ULN2001A-ULN2002A
ULN2003A-ULN2004A

SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT



DIP16

ORDERING NUMBERS: ULN2001A/2A/3A/4A

SO16

ORDERING NUMBERS: ULN2001D/2D/3D/4D

DESCRIPTION

The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

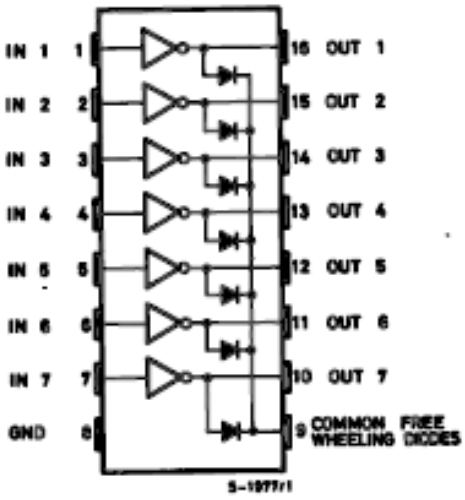
The four versions interface to all common logic families :

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6–15V CMOS, PMOS

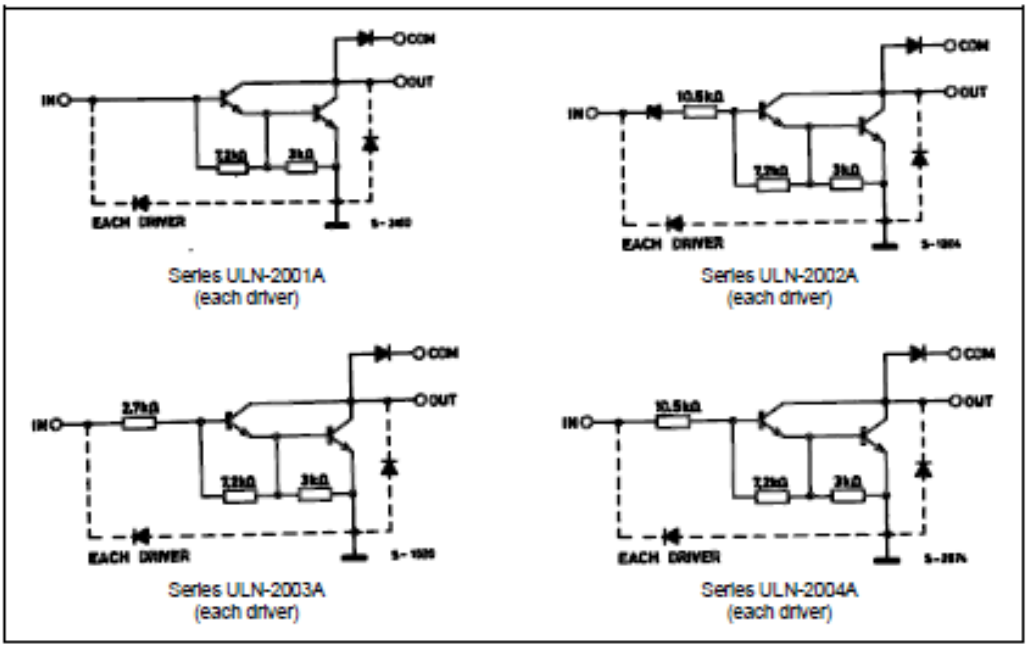
These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal printheads and high power buffers.

The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.

PIN CONNECTION



SCHEMATIC DIAGRAM



Series ULN-2001A (each driver)

Series ULN-2002A (each driver)

Series ULN-2003A (each driver)

Series ULN-2004A (each driver)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_{in}	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_c	Continuous Collector Current	500	mA
I_b	Continuous Base Current	25	mA
T_{amb}	Operating Ambient Temperature Range	– 20 to 85	°C
T_{stg}	Storage Temperature Range	– 55 to 150	°C
T_j	Junction Temperature	150	°C

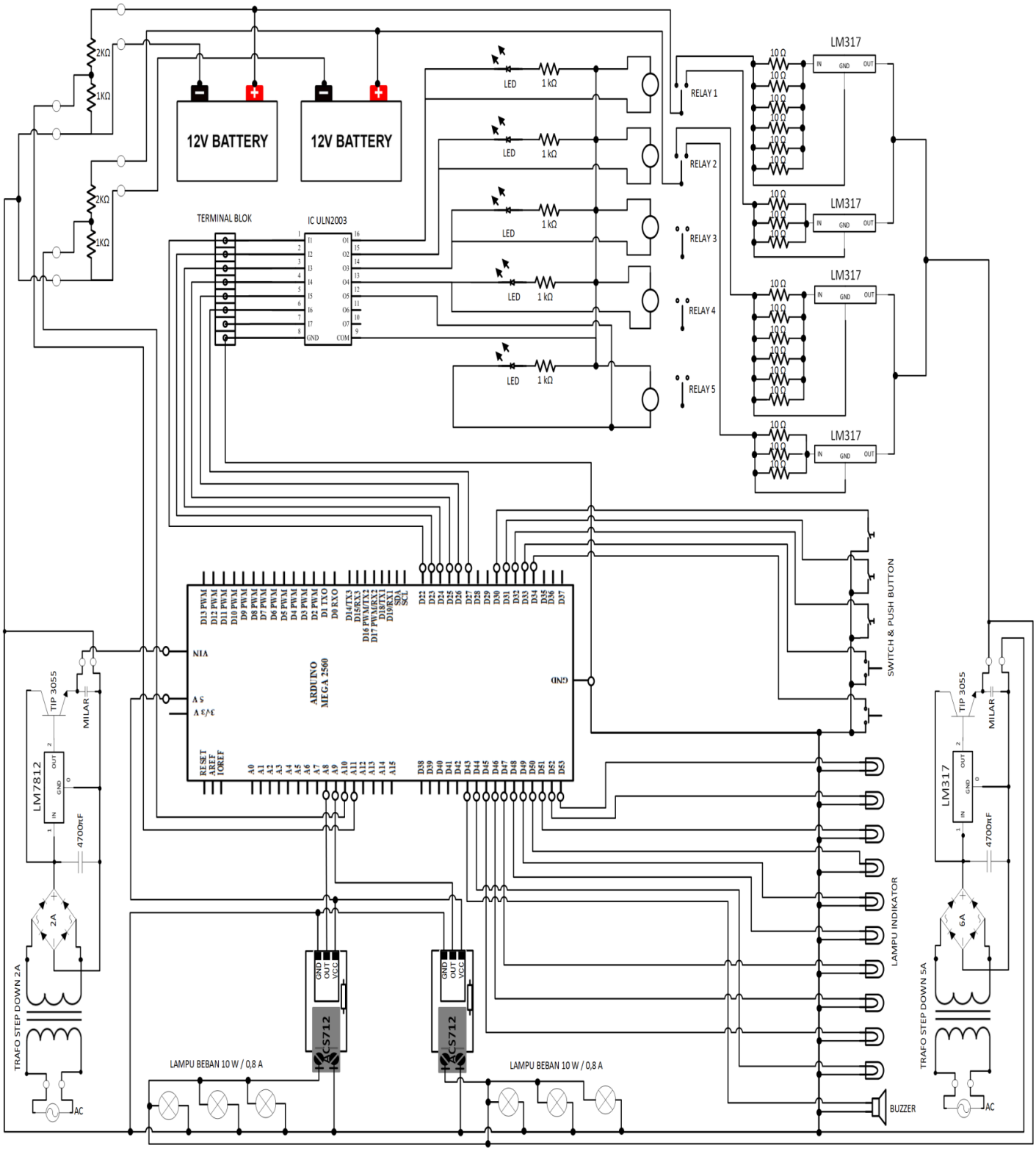
THERMAL DATA

Symbol	Parameter		DIP16	SO16	Unit
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	70	120	°C/W

ELECTRICAL CHARACTERISTICS (T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	V_{CE} = 50V			50	µA	1a
		T_{amb} = 70°C, V_{CE} = 50V			100	µA	1a
		T_{amb} = 70°C for ULN2002A V_{CE} = 50V, V_i = 6V			500	µA	1b
		for ULN2004A V_{CE} = 50V, V_i = 1V			500	µA	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	I_C = 100mA, I_B = 250µA		0.9	1.1	V	2
		I_C = 200 mA, I_B = 350µA		1.1	1.3	V	2
		I_C = 350mA, I_B = 500µA		1.3	1.6	V	2
$I_{i(on)}$	Input Current	for ULN2002A, V_i = 17V		0.82	1.25	mA	3
		for ULN2003A, V_i = 3.85V		0.93	1.35	mA	3
		for ULN2004A, V_i = 5V		0.35	0.5	mA	3
		V_i = 12V		1	1.45	mA	3
$I_{i(off)}$	Input Current	T_{amb} = 70°C, I_C = 500µA	50	65		µA	4
$V_{i(on)}$	Input Voltage	V_{CE} = 2V				V	5
		for ULN2002A I_C = 300mA			13		
		for ULN2003A I_C = 200mA			2.4		
		I_C = 250mA			2.7		
		I_C = 300mA			3		
		for ULN2004A I_C = 125mA			5		
		I_C = 200mA			6		
		I_C = 275mA			7		
		I_C = 350mA			8		
h_{FE}	DC Forward Current Gain	for ULN2001A V_{CE} = 2V, I_C = 350mA	1000				2
C_i	Input Capacitance			15	25	pF	
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	
I_R	Clamp Diode Leakage Current	V_R = 50V			50	µA	6
		T_{amb} = 70°C, V_R = 50V			100	µA	6
V_F	Clamp Diode Forward Voltage	I_F = 350mA		1.7	2	V	7

Lampiran 4 Rangkaian Keseluruhan Alat



Lampiran 5 Program Sistem Kerja Keseluruhan Alat

```
#include <TimerOne.h>
#include <Mudbus.h>
#include <SPI.h>
#include <Ethernet.h>
Mudbus Mb;

const int tegAwalKa = A10;
const int tegAwalKi = A11;
const int tegKa = A13;
const int tegKi = A12;
const int arusKi = A9;
const int arusKa = A8;
const int bateraiKa = A0;
const int LEDATS = 53;
const int bateraiKi = 25;
const int LEDDCFail = 49;
const int LEDACFail = 47;
const int LEDFloatKa = 33;
const int LEDBoostKa = 31;
const int LEDChargerKa = 29;
const int LEDACUtama = 29;
const int pushButtonAtas;
const int pushButtonBawah;
const int LEDFloatKi = 45;
const int LEDBoostKi = 43;
const int LEDChargerKi = 41;
const int LEDJalurUtama = 39;
const int tblrl = 40;
const int tbl1 = 22;
const int tbl2 = 24;
const int tbl3 = 26;
const int tblReset = 35;
const int tblHorn = 37;
const int tblRemote = 37;
const int RC1 = 30;
const int RC2 = 28;
const int ATS = 36;
const int RKi = 32;
```

```
const int RKa = 34;
const int RT = 38;
const int pinBuzzer = A15;

boolean statusATS = 1;
boolean statusBuzzer = 0;
boolean statusHorn = 0;
boolean sReset = 0;

int jumlahATS;
int jumlahKedip = 0;
int statusKa;
int statusKi;

float teganganKi;
float teganganKa;

const float bA = 13.5;
const float bB = 12.3;

unsigned long currentMillis;
unsigned long previousMillis = 0;
const long interval = 1000;

volatile boolean statusBateraiKi = 0;
volatile boolean statusBateraiKa = 0;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 20, 4);

void setup() {
  //SETTING IP ADDRESS MODBUS//
  uint8_t mac[]     = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // dirubah sesuai
kesepakatan
  uint8_t ip[]      = { 192, 168, 0, 123 };   // dirubah sesuai kesepakatan
  uint8_t gateway[] = { 192, 168, 0, 1 };
  uint8_t subnet[]  = { 255, 255, 255, 0 };
  Ethernet.begin(mac, ip, gateway, subnet);
  //Avoid pins 4,10,11,12,13 when using ethernet shield//
```

```
// put your setup code here, to run once:
pinMode(tbl1, INPUT);
pinMode(tbl2, INPUT);
pinMode(tbl3, INPUT);
pinMode(tblReset, INPUT);
pinMode(tblHorn, INPUT);
pinMode(tblrl, INPUT);

digitalWrite(tbl1, HIGH);
digitalWrite(tbl2, HIGH);
digitalWrite(tbl3, HIGH);
digitalWrite(tblReset, HIGH);
digitalWrite(tblHorn, HIGH);
digitalWrite(tblRemote, HIGH);
digitalWrite(tblrl, HIGH);

pinMode(ATS, OUTPUT);
pinMode(RKi, OUTPUT);
pinMode(RKa, OUTPUT);
pinMode(RT, OUTPUT);
pinMode(RC1, OUTPUT);
pinMode(RC2, OUTPUT);
pinMode(tblRemote, OUTPUT);
pinMode(LEDFloatKa, OUTPUT);
pinMode(LEDFloatKi, OUTPUT);
pinMode(LEDBoostKi, OUTPUT);
pinMode(LEDBoostKa, OUTPUT);
pinMode(LEDDCFail, OUTPUT);
pinMode(LEDACFail, OUTPUT);
pinMode(bateraiKi, OUTPUT);
pinMode(bateraiKa, OUTPUT);
pinMode(LEDChargerKa, OUTPUT);
pinMode(LEDChargerKi, OUTPUT);
pinMode(LEDJalurUtama, OUTPUT);
pinMode(pinBuzzer, OUTPUT);

//digitalWrite(pinBuzzer, HIGH);
//digitalWrite(bateraiKa, HIGH);
//digitalWrite(bateraiKi, HIGH);
//digitalWrite(ATS, HIGH);
```

```
  Serial.begin(9600);

  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(" ATS dan Monitoring");
  lcd.setCursor(0, 1);
  lcd.print("     Sistem DC");
  lcd.setCursor(0, 2);
  lcd.print("Oleh : ");
  lcd.setCursor(0, 3);
  lcd.print("Iwan dan Yesa");
  delay(1000);
  lcd.setCursor(0, 2);
  lcd.print("                ");
  lcd.setCursor(0, 3);
  lcd.print("                ");

  Timer1.initialize(500000); // set a timer of length 100000 microseconds (or 0.1 sec - or 10Hz => the led will blink 5 times, 5 cycles of on-and-off, per second)
  Timer1.attachInterrupt( kedip );
}

void loop() {
  Mb.Run();
  // put your main code here, to run repeatedly:
  bacaArusKi();
  bacaArusKa();
  bacaTeganganKa();
  bacaTeganganKi();
  bacaTeganganAwalKa();
  bacaTeganganAwalKi();
  if (digitalRead(tblrl) == LOW) {
    remote();
    ATSAktif2();
  } else {
    bacaTombol();
    ATSAktif();
  }
```

```
  charge();
  delay(100);
}

void bacaArusKi() {
  long ADC1 = 0;
  for (int i = 0; i < 60; i++) {
    ADC1 += analogRead(arusKi);
    delay(1);
  }
  float rata = ADC1 / 60.0;
  float V = rata * 5000 / 1023.0;
  float arus = (V - 2312) / 185;

  arus *= 1.5;

  if (arus < 0.15) {
    arus = 0;
  }
  if (arus > 2) {
    arus -= 0.2;
  }

  if (arus<1.0){
    arus+=0.18;
  }

  Mb.R[0] = int (arus * 100);

  Serial.println("ArusKi ");
  Serial.println(arus);
  Serial.println("  ");
  //Serial.println(arus);
  /*lcd.setCursor(0, 0);
  lcd.print("I:");
  lcd.print(arus);*/
}

void bacaArusKa() {
  long ADC1 = 0;
```

```
  for (int i = 0; i < 60; i++) {
    ADC1 += analogRead(arusKa);
    delay(1);
  }
  float rata = ADC1 / 60.0;
  float V = rata * 5000 / 1023.0;
  float arus = (V - 518) / 100;

  arus *= 0.83;

  if (arus < 0.15) {
    arus = 0;
  }

  if (arus > 2) {
    arus -= 0.3;
  }

  Mb.R[1] = int (arus * 100);

  Serial.println("ArusKa ");
  Serial.println(arus);
  Serial.println("  ");
  //Serial.println(arus);
  /*lcd.setCursor(10, 0);
  lcd.print("I2:");
  lcd.print(arus);*/

  //Serial.println(arus);
}

void bacaTeganganAwalKi() {
  long ADC1 = 0;
  for (int i = 0; i < 60; i++) {
    ADC1 += analogRead(tegAwalKi);
    delay(1);
  }
  float rata = ADC1 / 60.0;
  float V = rata * 5000 / 1023.0;
  float tegangan = V * 3 / 1000.0;
```

```
  if (tegangan > 9) {
    statusKi = 1;
    digitalWrite(LEDDCFail, LOW);
  } else {
    statusKi = 0;
    //digitalWrite(LEDDCFail, HIGH);
  }
  Mb.R[2] = int (tegangan * 100);

  Serial.println("TeganganAwalKi");
  Serial.println(tegangan);
  Serial.println("  ");
}

void bacaTeganganAwalKa() {
  long ADC1 = 0;
  for (int i = 0; i < 60; i++) {
    ADC1 += analogRead(tegAwalKa);
    delay(1);
  }
  float rata = ADC1 / 60.0;
  float V = rata * 5000 / 1023.0;
  float tegangan = V * 3 / 1000.0;


  if (tegangan > 9) {
    statusKa = 1;
    digitalWrite(LEDDCFail, LOW);
    Serial.println("Kanan");
  } else {
    statusKa = 0;
    Serial.println("Kanan mati");
    //digitalWrite(LEDDCFail, HIGH);
  }
  Mb.R[3] = int (tegangan * 100);

  Serial.println("TeganganAwalKa");
  Serial.println(tegangan);
  Serial.println("  ");
```

```
}

void bacaTeganganKi() {
  long ADC1 = 0;
  for (int i = 0; i < 60; i++) {
    ADC1 += analogRead(tegKi);
    delay(1);
  }
  float rata = ADC1 / 60.0;
  float V = rata * 5000 / 1023.0;
  teganganKi = V * 3 / 1000.0;

  teganganKi *= 1.05;

  if (teganganKi < 0.15) {
    teganganKi = 0;
  }

  if (teganganKi > 7 && teganganKi < 11.5) {
    teganganKi = 11.5;
  }

  if (teganganKi > 11.6 && teganganKi < 12.6) {
    teganganKi = 12.2;
  }
  Mb.R[4] = int (teganganKi * 100);

  Serial.println("TeganganKi");
  Serial.println(teganganKi);
  Serial.println("  ");
  lcd.setCursor(0, 3);
  lcd.print("V1:");
  lcd.print(teganganKi);

  //Serial.println(tegangan);
}

void bacaTeganganKa() {
  long ADC1 = 0;
  for (int i = 0; i < 60; i++) {
```

```
    ADC1 += analogRead(tegKa);
    delay(1);
  }
  float rata = ADC1 / 60.0;
  float V = rata * 5000 / 1023.0;
  teganganKa = V * 3 / 1000.0;

  teganganKa *= 1.03;;

  if (teganganKa < 0.15) {
    teganganKa = 0;
  }

  if (teganganKa > 7 && teganganKa < 11.5) {
    teganganKa = 11.5;
  }

  if (teganganKa > 11.6 && teganganKa < 12.6) {
    teganganKa = 12.2;
  }
  Mb.R[5] = int (teganganKa * 100);

  Serial.println("TeganganKa");
  Serial.println(teganganKa);
  Serial.println("  ");
  lcd.setCursor(10, 3);
  lcd.print("V2:");
  lcd.print(teganganKa);
  //Serial.println(tegangan);
}

void bacaTombol() {
  if (digitalRead(tbl1) == LOW) {
    digitalWrite(RKa, HIGH);
    digitalWrite(RKi, HIGH);
    digitalWrite(LEDACFail, HIGH);
    Mb.C[4] = HIGH;
    digitalWrite(LEDJalurUtama, LOW);
    statusBateraiKi = 1;
    statusBateraiKa = 1;
```

```
    statusBuzzer = 1;
  } else {
    digitalWrite(RKa, LOW);
    digitalWrite(RKi, LOW);
    digitalWrite(LEDJalurUtama, HIGH);
    digitalWrite(LEDACFail, LOW);
    Mb.C[4] = LOW;
    statusBateraiKi = 0;
    statusBateraiKa = 0;
    digitalWrite(bateraiKa, LOW);
    digitalWrite(bateraiKi, LOW);
    Mb.C[9] = LOW;
    Mb.C[10] = LOW;
  }
  if (digitalRead(tbl2) == LOW) {
    digitalWrite(RKa, HIGH);
    digitalWrite(LEDDCFail, HIGH);
    Mb.C[5] = HIGH;
    statusBuzzer = 1;
  } else {
    if (digitalRead(tbl1) == HIGH) {
      digitalWrite(RKa, LOW);
    }
  }
  if (digitalRead(tbl3) == LOW) {
    digitalWrite(RKi, HIGH);
    digitalWrite(LEDDCFail, HIGH);
    Mb.C[5] = HIGH;
    statusBuzzer = 1;
  } else {
    if (digitalRead(tbl1) == HIGH) {
      digitalWrite(RKi, LOW);
    }
  }
  tombolReset();
  if (digitalRead(tbl1) == HIGH && digitalRead(tbl2) == HIGH &&
digitalRead(tbl3) == HIGH) {
    statusBuzzer = 0;
    digitalWrite(pinBuzzer, LOW);
  }
```

```
  if (digitalRead(tbl2) == HIGH && digitalRead(tbl3) == HIGH) {
    digitalWrite(LEDDCFail, LOW);
    Mb.C[5] = LOW;
  }
  if (digitalRead(tblHorn) == LOW) {
    statusHorn = 1;
    digitalWrite(pinBuzzer, LOW);
  }
}

void remote() {
  if (Mb.C[0] == HIGH) {
    digitalWrite(RKa, HIGH);
    digitalWrite(RKi, HIGH);
    digitalWrite(LEDACFail, HIGH);
    //Mb.C[4] = HIGH;
    digitalWrite(LEDJalurUtama, LOW);
    statusBateraiKi = 1;
    statusBateraiKa = 1;
    statusBuzzer = 1;
  } if (Mb.C[0] == LOW) {
    digitalWrite(RKa, LOW);
    digitalWrite(RKi, LOW);
    digitalWrite(LEDJalurUtama, HIGH);
    digitalWrite(LEDACFail, LOW);
    //Mb.C[4] = LOW;
    statusBateraiKi = 0;
    statusBateraiKa = 0;
    digitalWrite(bateraiKa, LOW);
    digitalWrite(bateraiKi, LOW);
    //Mb.C[9] = LOW;
    //Mb.C[10] = LOW;
  }
  if (Mb.C[1] == HIGH) {
    digitalWrite(RKa, HIGH);
    digitalWrite(LEDDCFail, HIGH);
    Mb.C[5] = HIGH;
    statusBuzzer = 1;
  } if (Mb.C[1] == LOW) {
    if (Mb.C[0] == LOW) {
```

```
      digitalWrite(RKa, LOW);
    }
  }
  if (Mb.C[2] == HIGH) {
    digitalWrite(RKi, HIGH);
    digitalWrite(LEDDCFail, HIGH);
    Mb.C[5] = HIGH;
    statusBuzzer = 1;
  } if (Mb.C[2] == LOW) {
    if (Mb.C[0] == LOW) {
      digitalWrite(RKi, LOW);
    }
  }
  tombolReset();
  if (Mb.C[0] == LOW && Mb.C[1] == LOW && Mb.C[2] == LOW) {
    statusBuzzer = 0;
    digitalWrite(pinBuzzer, LOW);
  }
  if (Mb.C[1] == LOW && Mb.C[2] == LOW) {
    digitalWrite(LEDDCFail, LOW);
    Mb.C[5] = LOW;
  }
  if (digitalRead(tblHorn) == LOW) {
    statusHorn = 1;
    digitalWrite(pinBuzzer, LOW);
  }
}

void tombolReset() {
  if (digitalRead(tblReset) == LOW) {
    jumlahKedip = 0;
    resett();
  }
}

void resett() {
  sReset = 1;
  ATSmati();
  while (digitalRead(tbl2) == LOW) {
    if (jumlahKedip < 15) {
```

```
      digitalWrite(LEDChargerKa, HIGH);
      Mb.C[13] = HIGH;
      jumlahKedip++;
      delay(500);
      digitalWrite(LEDChargerKa, LOW);
      Mb.C[13] = LOW;
      delay(500);
    } else {
      ATSnyala();
    }
  }
  while (digitalRead(tbl3) == LOW) {
    if (jumlahKedip < 15) {
      digitalWrite(LEDChargerKi, HIGH);
      Mb.C[6] = HIGH;
      jumlahKedip++;
      delay(500);
      digitalWrite(LEDChargerKi, LOW);
      Mb.C[6] = LOW;
      delay(500);
    } else {
      ATSnyala();
    }
  }
  statusHorn = 0;
}

void ATSAktif() {
  lcd.setCursor(4, 2);
  lcd.print("ATS : ");
  lcd.print(jumlahATS);
  lcd.print(" kali");
  if (sReset == 0) {
    if ((digitalRead(tbl2) == LOW) || (digitalRead(tbl3) == LOW)) {
      digitalWrite(ATS, HIGH);
      digitalWrite(LEDATS, HIGH);
      if (statusATS == 1) {
        jumlahATS++;
        if (jumlahATS > 5) {
          jumlahATS = 1;
```

```
      }
      statusATS = 0;
     }
    } else {
     digitalWrite(ATS, LOW);
     digitalWrite(LEDATS, LOW);
     statusATS = 1;
    }
  }
  if ((digitalRead(tbl2) == HIGH) && (digitalRead(tbl3) == HIGH)) {
   sReset = 0;
  }
}

void ATSAktif2() {
  lcd.setCursor(4, 2);
  lcd.print("ATS : ");
  lcd.print(jumlahATS);
  lcd.print(" kali");
  if (sReset == 0) {
   if ((Mb.C[1] == HIGH) || (Mb.C[2] == HIGH)) {
     digitalWrite(ATS, HIGH);
     digitalWrite(LEDATS, HIGH);
     if (statusATS == 1) {
      jumlahATS++;
      if (jumlahATS > 5) {
       jumlahATS = 1;
      }
      statusATS = 0;
     }
    } else {
     digitalWrite(ATS, LOW);
     digitalWrite(LEDATS, LOW);
     statusATS = 1;
    }
  }
  if ((Mb.C[1] == LOW) || (Mb.C[2] == LOW)) {
   sReset = 0;
  }
}
```

```
void ATSmati() {
  digitalWrite(ATS, LOW);
  digitalWrite(LEDATS, LOW);
}
void ATSnyala() {
  digitalWrite(ATS, HIGH);
  digitalWrite(LEDATS, HIGH);
}

void charge() {
  if (statusKa == 1) {
    digitalWrite(LEDChargerKa, HIGH);
    Mb.C[13] = HIGH;
    if (teganganKa >= bB) {
      digitalWrite(RC1, HIGH);
      digitalWrite(LEDFloatKa, HIGH);
      digitalWrite(LEDBoostKa, LOW);
      Mb.C[12] = LOW;
      Mb.C[11] = HIGH;
    } else if (teganganKa < bB) {
      digitalWrite(RC1, LOW);
      digitalWrite(LEDBoostKa, HIGH);
      Mb.C[12] = HIGH;
      digitalWrite(LEDFloatKa, LOW);
      Mb.C[11] = LOW;
    }
    statusBateraiKa = 0;
  } else {
    //Serial.println("charger mati");
    digitalWrite(LEDChargerKa, LOW);
    Mb.C[13] = LOW;
    digitalWrite(LEDBoostKa, LOW);
    Mb.C[12] = LOW;
    digitalWrite(LEDFloatKa, LOW);
    Mb.C[11] = LOW;
  }
  if (statusKi == 1) {
    digitalWrite(LEDChargerKi, HIGH);
    Mb.C[6] = HIGH;
```

```
    if (teganganKi >= bB) {
      digitalWrite(RC2, HIGH);
      digitalWrite(LEDFloatKi, HIGH);
      digitalWrite(LEDBoostKi, LOW);
      Mb.C[7] = LOW;
      Mb.C[8] = HIGH;
    } else if (teganganKi < bB) {
      digitalWrite(RC2, LOW);
      digitalWrite(LEDBoostKi, HIGH);
      Mb.C[7] = HIGH;
      digitalWrite(LEDFloatKi, LOW);
      Mb.C[8] = LOW;
    }
    statusBateraiKi = 0;
  } else {
    digitalWrite(LEDChargerKi, LOW);
    Mb.C[6] = LOW;
    digitalWrite(LEDBoostKi, LOW);
    Mb.C[7] = LOW;
    digitalWrite(LEDFloatKi, LOW);
    Mb.C[8] = LOW;
  }
}

void kedip() {
  if (statusBateraiKa == 1) {
    digitalWrite(bateraiKa, digitalRead( bateraiKa ) ^ 1);
  }
  if (statusBateraiKi == 1) {
    digitalWrite(bateraiKi, digitalRead( bateraiKi ) ^ 1);
  }
  if (statusBuzzer == 1 && statusHorn == 0) {
    digitalWrite(pinBuzzer, digitalRead( pinBuzzer ) ^ 1);
  }
}
```