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.
### Microcontroller

- **Operating Voltage**: 5V
- **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

### ATmega2560

- **5V Input Voltage (recommended)**
- **7-12V Input Voltage**

---

**Diagram**: Arduino Mega 2560 layout with labeled parts such as TX/RX Leds, digital pins, Power Led, USB Interface, External Power Supply, ICSP Header, ATmega 2560, and Reset Button.
The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the analogWrite() function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I2C:** 20 (SDA) and 21 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as
the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.
The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The ATmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.
Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).
How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You’ll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. http://arduino.cc/en/Guide/HomePage

Once you have downloaded/unzipped the arduino IDE, you can plug the Arduino to your PC via USB cable.

Linux Install  Windows Install  Mac Install

Now you’re actually ready to “burn” your first program on the arduino board. To select “blink led”, the physical translation of the well known programming “hello world”, select

File>Sketchbook>Arduino-0017>Examples>Digital>Blink

Once you have your sketch you’ll see something very close to the screenshot on the right.

In Tools>Board
select MEGA

Now you have to go to

Tools> Serial Port

and select the right serial port, the one arduino is attached to.
1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS," THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each
Arduino board produced is created / looked after half squared Km of Costa Rica’s forest’s.
LAMPIRAN B1 DataSheet LCD

20 x 4 Character LCD

FEATURES
- Type: Character
- Display format: 20 x 4 characters
- Built-in controller: ST 7066 (or equivalent)
- Duty cycle: 1/16
- 5 x 8 dots includes cursor
- + 5 V power supply (also available for + 3 V)
- LED can be driven by pin 1, pin 2, pin 15, pin 16 or A and K
- N.V. optional for + 3 V power supply
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912

ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SYMBOL</th>
<th>STANDARD VALUE</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply</td>
<td>V_{DD} to V_{SS}</td>
<td>-0.3</td>
<td>7.0 V</td>
</tr>
<tr>
<td>Input Voltage</td>
<td>V_i</td>
<td>-0.3</td>
<td>V_{DD}</td>
</tr>
</tbody>
</table>

Note
- V_{SS} = 0 V, V_{DD} = 5.0 V

ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SYMBOL</th>
<th>CONDITION</th>
<th>STANDARD VALUE</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage</td>
<td>V_{DD}</td>
<td>V_{DD} = + 5 V</td>
<td>4.7</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V_{DD} = + 3 V</td>
<td>2.7</td>
<td>3.0</td>
</tr>
<tr>
<td>Supply Current</td>
<td>I_{DD}</td>
<td>V_{DD} = + 5 V</td>
<td>-</td>
<td>8.0</td>
</tr>
<tr>
<td>Recommended LC Driving Voltage for Normal Temperature Version Module</td>
<td>V_{DD} to V_{0}</td>
<td>-20 °C</td>
<td>5.0</td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 °C</td>
<td>4.6</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25 °C</td>
<td>4.1</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50 °C</td>
<td>3.9</td>
<td>4.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70 °C</td>
<td>3.7</td>
<td>3.9</td>
</tr>
<tr>
<td>LED Forward Voltage</td>
<td>V_{F}</td>
<td>25 °C</td>
<td>-</td>
<td>4.2</td>
</tr>
<tr>
<td>LED Forward Current</td>
<td>I_{F}</td>
<td>25 °C</td>
<td>-</td>
<td>540</td>
</tr>
<tr>
<td>EL Power Supply Current</td>
<td>I_{EL}</td>
<td>V_{EL} = 110 V_{AC}, 400 Hz</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

OPTIONS

<table>
<thead>
<tr>
<th>PROCESS COLOR</th>
<th>BACKLIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TN</td>
<td>STN Gray</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
For detailed information, please see the “Product Numbering System” document.

### DISPLAY CHARACTER ADDRESS CODE

<table>
<thead>
<tr>
<th>Display Position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD RAM Address</td>
<td>00</td>
<td>01</td>
<td>02</td>
<td>03</td>
<td>04</td>
<td>05</td>
<td>06</td>
<td>07</td>
<td>08</td>
<td>09</td>
<td>0A</td>
<td>0B</td>
<td>0C</td>
<td>0D</td>
<td>0E</td>
<td>0F</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>DD RAM Address</td>
<td>01</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>4A</td>
<td>4B</td>
<td>4C</td>
<td>4D</td>
<td>4E</td>
<td>4F</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
</tr>
<tr>
<td>DD RAM Address</td>
<td>04</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>1A</td>
<td>1B</td>
<td>1C</td>
<td>1D</td>
<td>1E</td>
<td>1F</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>DD RAM Address</td>
<td>05</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>5A</td>
<td>5B</td>
<td>5C</td>
<td>5D</td>
<td>5E</td>
<td>5F</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
</tr>
</tbody>
</table>

### INTERFACE PIN FUNCTION

<table>
<thead>
<tr>
<th>PIN NO.</th>
<th>SYMBOL</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V_{SS}</td>
<td>Ground</td>
</tr>
<tr>
<td>2</td>
<td>V_{DD}</td>
<td>+3 V or +5 V</td>
</tr>
<tr>
<td>3</td>
<td>V_{O}</td>
<td>Contrast adjustment</td>
</tr>
<tr>
<td>4</td>
<td>RS</td>
<td>H/L register select signal</td>
</tr>
<tr>
<td>5</td>
<td>R/W</td>
<td>H/L read/write signal</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
<td>H → L enable signal</td>
</tr>
<tr>
<td>7</td>
<td>DB0</td>
<td>H/L data bus line</td>
</tr>
<tr>
<td>8</td>
<td>DB1</td>
<td>H/L data bus line</td>
</tr>
<tr>
<td>9</td>
<td>DB2</td>
<td>H/L data bus line</td>
</tr>
<tr>
<td>10</td>
<td>DB3</td>
<td>H/L data bus line</td>
</tr>
<tr>
<td>11</td>
<td>DB4</td>
<td>H/L data bus line</td>
</tr>
<tr>
<td>12</td>
<td>DB5</td>
<td>H/L data bus line</td>
</tr>
<tr>
<td>13</td>
<td>DB6</td>
<td>H/L data bus line</td>
</tr>
<tr>
<td>14</td>
<td>DB7</td>
<td>H/L data bus line</td>
</tr>
<tr>
<td>15</td>
<td>A</td>
<td>Power supply for LED (4.2 V)</td>
</tr>
<tr>
<td>16</td>
<td>K</td>
<td>Power supply for B/L (0 V)</td>
</tr>
<tr>
<td>17</td>
<td>NC/VE</td>
<td>NC or negative voltage output</td>
</tr>
<tr>
<td>18</td>
<td>NC</td>
<td>NC connection</td>
</tr>
</tbody>
</table>
ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, “Vishay”), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay’s knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer’s responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and / or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer’s technical experts. Product specifications do not expand or otherwise modify Vishay’s terms and conditions of purchase, including but not limited to the warranty expressed therein.

Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.
#include <LiquidCrystal.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

Servo motorKeluar;
Servo motorMasuk;

LiquidCrystal_I2C lcd(0x27, 20, 4);

const int Masuk = 31;
const int sensor1 = 2;
const int sensor2 = 3;
const int sensor3 = 4;
const int sensor4 = 5;
const int sensor5 = 6;
const int sensor6 = 7;
const int Keluar = 12;
const int BZR = 10;
const int LED1 = 11;

int siap=0;
int total=0;
int s1=0;
int s2=0;
int s3=0;
int s4=0;
int s5=0;
int s6=0;

int cup1=0;
int cup2=0;
void setup() {
    Serial.begin(9600);
    lcd.begin();
lcd.begin();
lcd.backlight();
lcd.setCursor(3,0);
    lcd.print("SELAMAT DATANG");
    delay(2000);
lcd.setCursor(1,2);
lcd.print(" DEDE PARKING AREA");
delay(2000);
lcd.clear();

motorKeluar.attach(A0);
motorMasuk.attach(A1);

pinMode(Masuk, INPUT);
pinMode(sensor1, INPUT);
pinMode(sensor2, INPUT);
pinMode(sensor3, INPUT);
pinMode(sensor4, INPUT);
pinMode(sensor5, INPUT);
pinMode(sensor6, INPUT);
pinMode(Keluar, INPUT);
pinMode(BZR, OUTPUT);
pinMode(LED1, OUTPUT);

void loop() {
    bacaMasuk = digitalRead(Masuk);
    bacaKeluar = digitalRead(Keluar);

    scan();

    if (bacaKeluar == 0){
        motorKeluar.write(90);
        delay(1000);
    }
}
else {
    motorKeluar.write(0);
}

//=================================
if (bacaMasuk == 0 && total<6) {
    //bacaParkir?
    String cetak="";
    if(s1==0 && siap==0) {cetak="P1";cup1=1;siap=1; Serial.print("@"); Serial.println(cetak); } 
    else if(s2==0 && siap==0) {cetak="P2";cup2=1;siap=1;Serial.print("@"); Serial.println(cetak); } 
    else if(s3==0 && siap==0) {cetak="P3";cup3=1;siap=1;Serial.print("@"); Serial.println(cetak); } 
    else if(s4==0 && siap==0) {cetak="P4";cup4=1;siap=1;Serial.print("@"); Serial.println(cetak); } 
    else if(s5==0 && siap==0) {cetak="P5";cup5=1;siap=1;Serial.print("@"); Serial.println(cetak); } 
    else if(s6==0 && siap==0) {cetak="P6";cup6=1;siap=1;Serial.print("@"); Serial.println(cetak); } 
    motorMasuk.write(0);
}
else if (bacaMasuk == 0 && total==6) {
    motorMasuk.write(90);
    lcd.clear();
    lcd.setCursor(6,1);
    lcd.print("PARKIRAN");
    lcd.setCursor(8,2);
    lcd.print("PENUH");
    bunyi();
    siap=0;
}
else {

siap=0;
motorMasuk.write(90);
delay(1000);
}

//=================================

delay(1000);
}

void scan(){
    String status1="KOSONG";
    String status2="KOSONG";
    String status3="KOSONG";
    String status4="KOSONG";
    String status5="KOSONG";
    String status6="KOSONG";
    s1=0;
    s2=0;
    s3=0;
    s4=0;
    s5=0;
    s6=0;
    bacaSensor1 = digitalRead(sensor1);
    bacaSensor2 = digitalRead(sensor2);
    bacaSensor3 = digitalRead(sensor3);
    bacaSensor4 = digitalRead(sensor4);
    bacaSensor5 = digitalRead(sensor5);
    bacaSensor6 = digitalRead(sensor6);
    if(bacaSensor1 == 0){
s1=1;
status1="TERISI";
sudah1=1;
}
else if(bacaSensor1 == 1 && sudah1==1){
cup1=0;
sudah1=0;
}
else if(cup1==1){
s1=1;
status1="TERISI";
}

//==================================================

if(bacaSensor2 == 0){
s2=1;
sudah2=1;
status2="TERISI";
}
else if(bacaSensor2 == 1 && sudah2==1){
cup2=0;
sudah2=0;
}
else if(cup2==1){
s2=1;
status2="TERISI";
}

==================================

if(bacaSensor3 == 0){
s3=1;
sudah3=1;
status3="TERISI ";
}
else if(bacaSensor3 == 1 && sudah3==1) {
cup3=0;
sudah3=0;
}
else if(cup3==1) {
s3=1;
    status3="TERISI ";
}
//==================================================
if(bacaSensor4 == 0){
s4=1;
sudah4=1;
status4="TERISI ";
}
else if(bacaSensor4 == 1 && sudah4==1) {
cup4=0;
sudah4=0;
}
else if(cup4==1) {
s4=1;
    status4="TERISI ";
}
//==================================================
if(bacaSensor5 == 0){
s5=1;
sudah5=1;
status5="TERISI ";
}
else if(bacaSensor5 == 1 && sudah5==1) {
cup5=0;
sudah5=0;
}
else if(cup5==1){
s5=1;
    status5="TERISI ";
}
//==================================================
if(bacaSensor6 == 0){
s6=1;
sudah6=1;
    status6="TERISI ";
}
else if(bacaSensor6 == 1 & sudah6==1){
cup6=0;
sudah6=0;
}
else if(cup6==1){
s6=1;
    status6="TERISI ";
}
total=s1+s2+s3+s4+s5+s6;
    statusparkir="= PARKIR :TERSEDIA =";
if(total==6){statusparkir="STATUS PARKIR PENUH=";}

    lcd.setCursor(0,0);lcd.print("P1");
lcd.setCursor(3,0);lcd.print(status1);
lcd.setCursor(11,0);lcd.print("P2");
lcd.setCursor(14,0);lcd.print(status2);
lcd.setCursor(0,1);lcd.print("P3");
lcd.setCursor(3,1);lcd.print(status3);
lcd.setCursor(11,1);lcd.print("P4 ");
lcd.setCursor(14,1);lcd.print(status4);
lcd.setCursor(0,2);lcd.print("P5");
lcd.setCursor(3,2);lcd.print(status5);
lcd.setCursor(11,2);lcd.print("P6");
lcd.setCursor(14,2);lcd.print(status6);
lcd.setCursor(0,3);
lcd.print(statusparkir);

Serial.print(s1);
Serial.print("#");
Serial.print(s2);
Serial.print("#");
Serial.print(s3);
Serial.print("#");
Serial.print(s4);
Serial.print("#");
Serial.print(s5);
Serial.print("#");
Serial.print(s6);
Serial.println("#");
}

void bunyi() {
  tone(BZR,500,500);
  digitalWrite(LED1,HIGH);
  delay(1000);
  noTone(BZR);
  digitalWrite(LED1,LOW);
}
LAMPIRAN E1 Table Komponen Harga dan Alat

<table>
<thead>
<tr>
<th>NOMER</th>
<th>KOMPONEN</th>
<th>JUMLAH</th>
<th>HARGA SATUAN</th>
<th>HARGA</th>
<th>TOTAL HARGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Arduino Mega</td>
<td>1</td>
<td>Rp.140.000</td>
<td>Rp.140.000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Servo</td>
<td>4</td>
<td>Rp. 16.000</td>
<td>Rp.72.000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>LCD</td>
<td>2</td>
<td>Rp. 75.000</td>
<td>Rp.150.000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Infrared</td>
<td>12</td>
<td>Rp. 7000</td>
<td>Rp.84.000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Kabel Jumper</td>
<td>6</td>
<td>Rp. 15.000</td>
<td>Rp.90.000</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Breaboard</td>
<td>1</td>
<td>Rp.13.000</td>
<td>Rp.13.000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Harga Keseluruhan</td>
<td></td>
<td></td>
<td></td>
<td>Rp. 549.000</td>
</tr>
</tbody>
</table>

Harga Keseluruhan: Rp. 549.000
FOTO ALAT LAMPIRAN F4