

Home Secrity Using Fingerprint And Image Processing

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology degree in

Information Technology by

Keerthivasan.E (37120040)

Jagannathan . S (37120033)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND
TECHNOLOGY**

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI -

600 119

MARCH – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344
www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of E.Keerthivasan(37120040) and who carried out the project entitled "Home Security Using Fingerprint And Image Processing" under our supervision from November 2020 to March 2021.

Internal Guide

Ms . T.Anandhi, M.E.,

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voice Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I am E.Keerthivasan and hereby declare that the project report entitled on **Home Secrity Using Fingerprint And Image Processing** done by me under the guidance of **Ms.T.Anandhi,M.E**(internal) is submitted in partial fulfillment of the requirements for the Award of Bachelor of Engineering/Technology degree in **INFORMATION TECHNOLOGY**

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr.R.Subhashini M.E., Ph.D., Head of the Department** of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project Guide **T.Anandhi M.E**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project

CHAPTER NO	TITLE	PAGE NO
1	INTRODUCTION	
2	LITERATURE SURVEY	
	2.1 Literature Survey	
	2.2 Review of the Literature Survey	
3.	AIM AND SCOPE	
	3.1 Aim	
	3.2 Scope	
	3.3 Existing system	
4	METHODOLOGY	
	4.1 Hardware and Software Requirements	
	4.2 working principle	
5	ARDUINO MICRO CONTROLLER	
	5.1 Introduction to Arduino	
	5.2 Arduino mega	
	5.3 Memory	
	5.4 Input and Output	
	5.5 Communication	
	5.6 Application	
	5.7 GSM modem	
	5.8 Servo motor	
6	RESuLT AND DISCUSSION	
	6.1 Result and Discussion	
7	CONCLUSION AND FUTURE WORK	
	7.1 Conclusion	

	7.2 Future work	
8	REFERENCE	
9	SAMPLE CODE	

CHAPTER 1

INTRODUCTION

In today's world, there is a high risk of losing a data because of low level security system. To avoid this kind of problems, we have come along with an idea that improve security system, Our Security system having two level securitys, FINGERPRINTS, FACE RECOGNITION. If you have only one security it is very easy to open. So we use two level of system when you have two level. It's very hard to open. I am sure my project very help to Peoples for keeping their things safe.

CHAPTER 2

LITERATURE SURVEY

2.1Literatre Survey

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis

2.2 Review of the literature review

TITLE: Android & Bluetooth Module Based Door Automation System

AUTHOR: Harshit Singhal

The most popular home controllers are those that are connected to a Linux, Windows based PC and android mobile. In our research we presented a part of secure home technology which using Bluetooth in a mobile device, so it will more easy and efficient to use. It also based on Android and Arduino (ATmega 328) platform both of which are free open source software. In this paper, a system called door locks automation system using Bluetooth-based Android and IOS Smartphone is proposed and prototyped.

TITLE: SMART DOOR LOCKING SYSTEM

AUTHOR:NISHAD

A smart phone is now a very common device that everyone carries with them all the time. Smart phones have enabled us to perform various tasks while using a single device.

TITLE: Design and Analysis of IOT Based Real Time System for Door Locking/Unlocking using Face Identification

AUTHOR: Ajitesh Kumar

. So for resolving this issue we are proposes a real time application controlled door locking/unlocking mechanism which harnesses the power of IOT and machine learning for smooth functionality. The door unlocking system proposed

here uses a Raspberry Pi 3 model B for computation along with a Pi Camera to take face as an input of the user

TITLE: Design and Implementation of Password Based Security Lock System

AUTHOR: M. Hymavathi

The security of any organization or house is of prime importance always. The concern is for the physical property and also for the intellectual property in case of any organization. This proposed system provides a user friendly security system for organizations and homes. This system is password based and allows only authorized person to access it with a password.

TITLE: IoT based facial recognition door access control home security system using raspberry pi

AUTHOR: A. R. Syafeeza

The home security system has become vital for every house. Previously, most doors can be open by using traditional ways, such as keys, security cards, password or pattern. However, incidents such as a key loss has led to much worrying cases such as robbery and identity fraud

CHAPTER 3

AIM AND SCOPE

3.1 Aim

The aim of the project is to provide security to the house from thief. This ensures additional security to the house . Without the access no one can open the door.

3.2 Scope

IT is mainly intended to automated security access operations using a mobile phone. The mobile phone present in the **system** uses auto answer function to lift the call. Each key in the mobile phone transmits two tones with different frequencies when pressed.

3.3 Existing system

In existing system, home security is not implemented because there is no sensor to perform automatic work

Drawbacks of existing system

- Door lock can be broken easily by intruder.

No alert system

Proposed System

In proposed system, home automation is implemented with the help of sensor to perform automatic work.

Advantages of proposed system

- Door lock cannot be broken.

Automated alert system

CHAPTER 4

METHODOLOGY

4.1 Hardware and Software requirements

Hardware requirements

- ARDUINO MEGA
- KEYPAD
- LCD
- SERVO MOTOR
- FINGERPRINT MODULE
- WIRED USB WEB CAMERA
- GSM
- TTL
- KEYPAD

Software requirements

- ARDUINO IDE
- EMBEDDED C
- MATLAB

4.2 Working principle

In this system, we have Arduino MEGA microcontroller which acts as brain of our system; hence the entire system program is stored in it. The fingerprint module is used to monitor the registered fingerprint and the wired camera is used to match the registered faces once this process is properly done the door mechanism is opened. In case the finger print or face match are not matched this system sends a alert message to owner. The keypad is used to enter the OTP to access the door mechanism. The information is displayed in LCD.

CHAPTER 5

ARDUINO MICRO CONTROLLER

5.1 Introduction to Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a

motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific

instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50.
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing

programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

5.2 ARDUINO MEGA:

The MEGA 2560 is designed for more complex projects. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch it is the recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities.

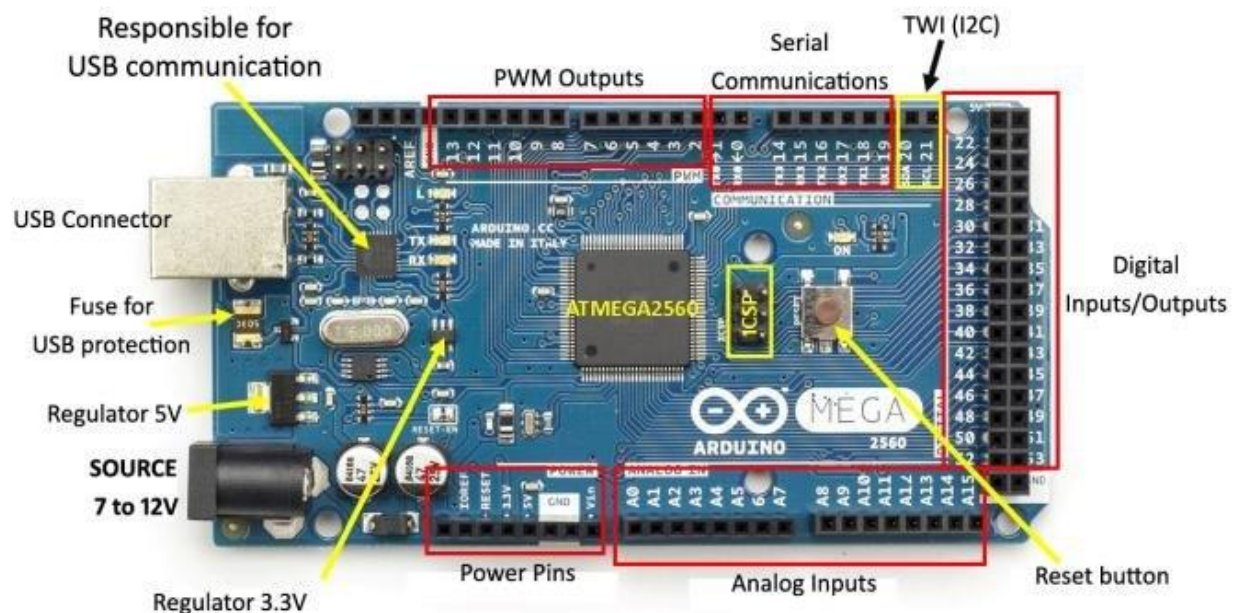


The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

TECHNICAL SPECIFICATIONS:

Microcontroller	<u>ATmega2560</u>
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by boot loader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

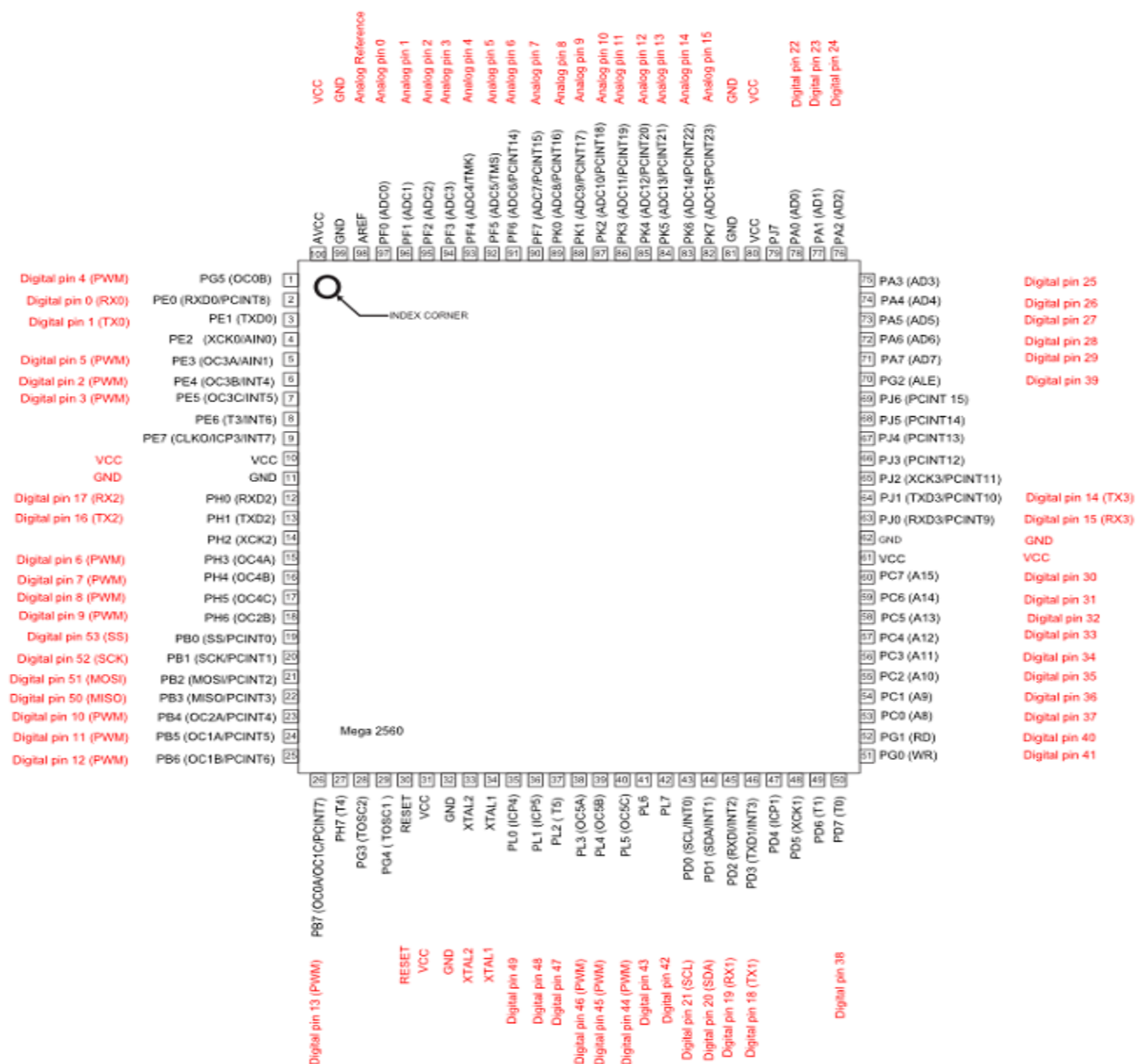


5.4 Input and Output

ATmega2560-Arduino Pin Mapping.

Below is the pin mapping for the Atmega2560. The chip used in Arduino 2560. There are pin mappings to Atmega8 and Atmega 168/328 as well.

Arduino Mega 2560 PIN diagram



Arduino Mega 2560 PIN mapping table

Pin Number	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLK0/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)

24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	

50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25

76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

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 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.

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 ATmega16U2 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 level, a rising or falling edge, or a change in level. See the attachInterrupt() function for details.
- **PWM:** 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analogWrite() function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino /Genuino Uno and the old Duemilanove and Diecimila Arduino boards.

- **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.
- **TWI:** 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards.

See also the mapping Arduino Mega 2560 PIN diagram.

The Mega 2560 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 is it 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.

COMMUNICATION:

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (ATmega 8U2 on the revision 1 and revision 2 boards) 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 (IDE) 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/ATmega16U2 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 2560's digital pins.

The Mega 2560 also supports TWI and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.

PHYSICAL CHARACTERISTICS AND SHIELD COMPATIBILITY:

The maximum length and width of the Mega 2560 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 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. 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. Furthermore, 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 2560 and Duemilanove / Diecimila boards. Please note that I2C is not located on the same pins on the Mega 2560 board (20 and 21) as the Duemilanove / Diecimila boards (analog inputs 4 and 5).

AUTOMATIC (SOFTWARE) RESET:

Rather than requiring a physical press of the reset button before an upload, the Mega 2560 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 (IDE) 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 Mega 2560 board 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 ATmega2560. 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 2560 board 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.

REVISIONS:

The Mega 2560 does not use the FTDI USB-to-serial driver chip used in past designs. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 Arduino boards) programmed as a USB-to-serial converter. Revision 2 of the Mega 2560 board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode](#). Revision 3 of the Arduino board and the current Genuino Mega 2560 have the following improved features:

- **1.0 pinout:** SDA and SCL pins - near to the AREF pin - and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the board that uses ATSAM3X8E, that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.

- Atmega 16U2 replace the 8U2.

APPLICATIONS:

- Arduboy, a handheld game console based on Arduino.
- Arduinome, a MIDI controller device that mimics the Monome.
- Ardupilot, drone software and hardware.
- ArduSat, a cubesat based on Arduino.
- C-STEM Studio, a platform for hands-on integrated learning of computing, science, technology, engineering, and mathematics (C-STEM) with robotics.
- Data loggers for scientific research.
- OBduino, a trip computer that uses the on-board diagnostics interface found in most modern cars.

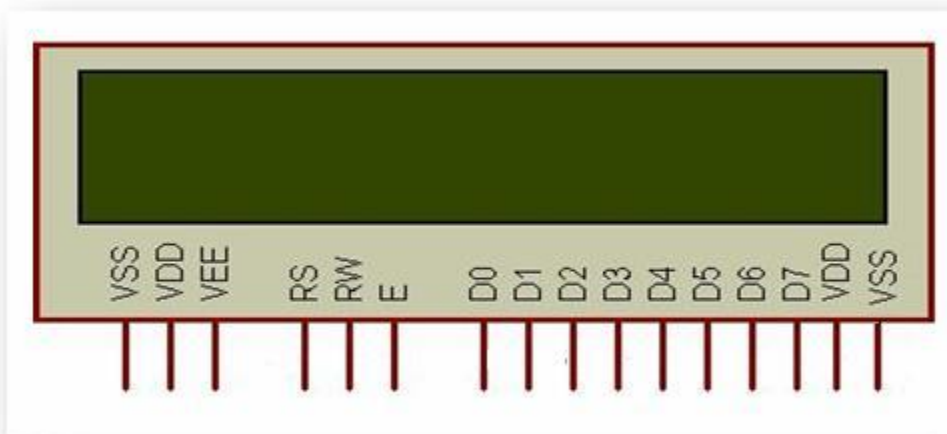
LCD (Liquid Crystal Display)

LCD screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

Pin Diagram:



Pin Description:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	V _{cc}
3	Contrast adjustment; through a variable resistor	V _{EE}

4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V _{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Introduction

The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. In this tutorial, we will discuss about character based LCDs, their interfacing with various microcontrollers, various interfaces (8-bit/4-bit), programming, special stuff and tricks you can do with these simple looking LCDs which can give a new look to your application.

Pin Description

The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, whereas LCDs supporting more than 80 characters make use of 2 HD44780 controllers.

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections). Pin description is shown in the table below.

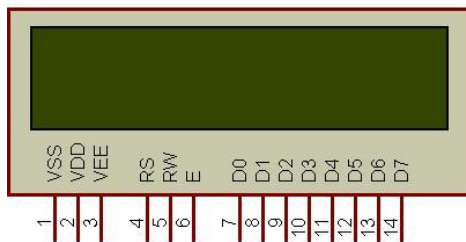


Figure 1: Character LCD type HD44780 Pin diagram

Pin No.	Name	Description
---------	------	-------------

Pin no. 1	D7	Data bus line 7 (MSB)
Pin no. 2	D6	Data bus line 6
Pin no. 3	D5	Data bus line 5
Pin no. 4	D4	Data bus line 4
Pin no. 5	D3	Data bus line 3
Pin no. 6	D2	Data bus line 2
Pin no. 7	D1	Data bus line 1
Pin no. 8	D0	Data bus line 0 (LSB)
Pin no. 9	EN1	Enable signal for row 0 and 1 (1 st controller)
Pin no. 10	R/W	0 = Write to LCD module 1 = Read from LCD module
Pin no. 11	RS	0 = Instruction input 1 = Data input
Pin no. 12	VEE	Contrast adjust
Pin no. 13	VSS	Power supply (GND)
Pin no. 14	VCC	Power supply (+5V)
Pin no. 15	EN2	Enable signal for row 2 and 3 (2 nd controller)
Pin no. 16	NC	Not Connected

Table 2: Character LCD pins with 2 Controller

Usually these days you will find single controller LCD modules are used more in the market. So in the tutorial we will discuss more about the single controller LCD,

the operation and everything else is same for the double controller too. Lets take a look at the basic information which is there in every LCD.

BF - Busy Flag:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing an exact amount of delay for the LCD processing. To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

Instruction Register (IR) and Data Register (DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. when send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD. We will discuss more on LCD instruction set further in this tutorial.

Commands and Instruction set

Only the instruction register (IR) and the data register (DR) of the LCD can be controlled by the MCU. Before starting the internal operation of the LCD, control information is temporarily stored into these registers to allow interfacing with various MCUs, which operate at different speeds, or various peripheral control devices. The internal operation of the LCD is determined by signals sent from the MCU. These signals, which include register selection signal (RS), read/write signal (R/W), and the data bus (DB0 to DB7), make up the LCD instructions (Table 3). There are four categories of instructions that:

- Designate LCD functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions

Although looking at the table you can make your own commands and test them. Below is a brief list of useful commands which are used frequently while working on the LCD.

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32

4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add*	128+add*
17	Set CGRAM address or set pointer	0x40+add**	64+add**

to CGRAM location

Table 4: Frequently used commands and instructions for LCD

* DDRAM address given in LCD basics section see Figure 2,3,4

** CGRAM address from 0x00 to 0x3F, 0x00 to 0x07 for char1 and so on.

LCD LIQUID CRYSTAL DISPLAYS:

The LCD standard requires 3 control lines and 8 I/O lines for the data bus.

- **8 data pins D7:D0**

Bi-directional data/command pins.

Alphanumeric characters are sent in ASCII format.

- **RS: Register Select**

RS = 0 -> Command Register is selected

RS = 1 -> Data Register is selected

- **R/W: Read or Write**

0 -> Write, 1 -> Read

- **E: Enable (Latch data)**

The 8 data lines are connected to PORT 1 of 8051 microcontroller. The three control lines (RS, RW and EN) are connected to PORT 3.5,3.6 and 3.7 respectively.

5.7 GSM Modem

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves. The working of GSM modem is based on commands, the commands always start with AT (which means ATtention) and finish with a <CR> character. For example, the dialing command is ATD<number>; ATD3314629080; here the dialing command ends with semicolon.

The AT commands are given to the GSM modem with the help of PC or controller. The GSM modem is serially interfaced with the controller with the help of MAX 232. Here max 232 acts as driver which converts TTL levels to the RS 232 levels. For serial interface GSM modem requires the signal based on RS 232 levels. The T1_OUT and R1_IN pin of MAX 232 is connected to the TX and RX pin of GSM modem



1.1 OBJECT OF THE DOCUMENT

This document gives an overview of the LinkSprite GSM/GPRS module: a miniature, single-side board, quad-band GSM 850/EGSM 900/DCS 1800/PCS 1900 module, ready for integration in various kinds of Fix wireless phones and other wireless devices.

1.2 SYSTEM BLOCK DIAGRAM

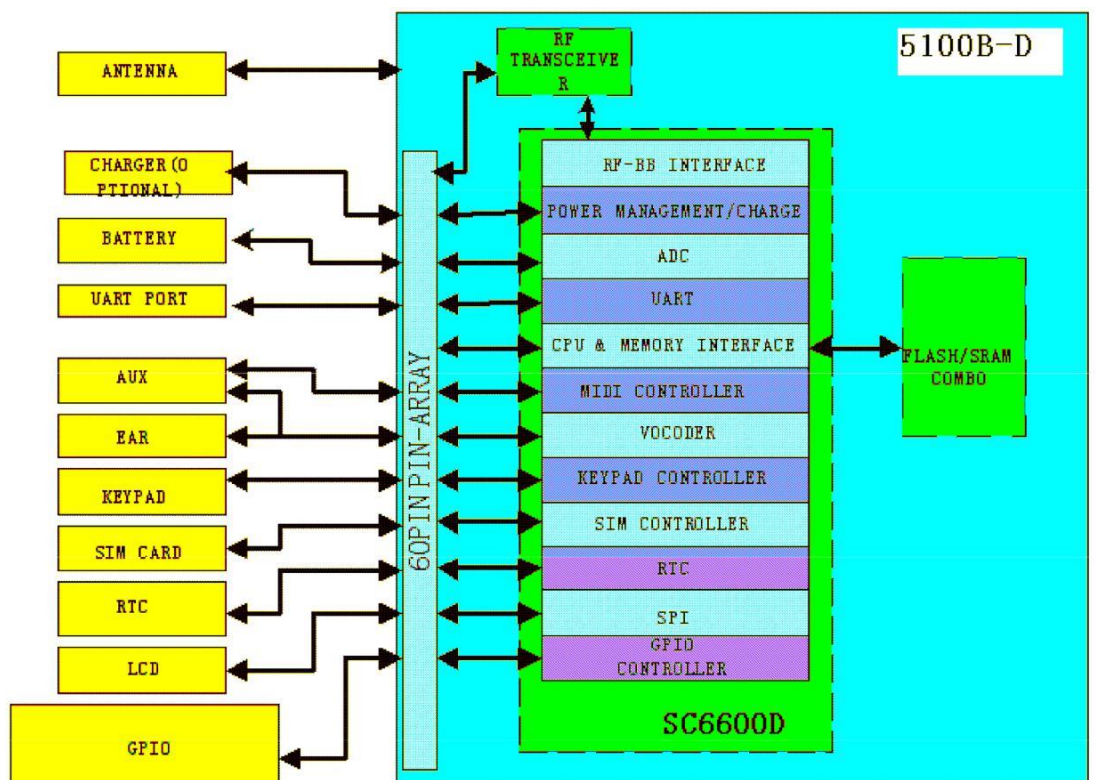


Figure 1: System block diagram

1.3 PRODUCT OVERVIEW

Temperature range	Normal range: -10°C to +55°C (full compliant)
	Storage: -40°C to +85°C
Weight	< 9g
Physical dimensions	35.0X39.0X2.9 mm (typical)
Connection	60 pins
Power supply	VBAT: 3.3V to 4.2V range, 3.6V typical.

Power consumption	Off mode: <100uA Sleep mode: <2.0mA Idle mode: <7.0mA (average)
	Communication mode: 350 mA (average,GSM) Communication mode: 2000mA (Typical peak during TX slot,GSM)
Li-ion Battery charging management and interface (OPTION)	Li-ion Battery charging management is included. The charger interface is provided on 60-pin connector. (only for 3.7V Li-ion Battery)
Frequency bands	EGSM900 +GSM850+ DCS1800+PCS1900
Transmit power	Class 4 (2W) for EGSM900/GSM850 Class 1 (1W) for DCS1800/PCS1900
Supported SIM card	3V/1.8V SIM card. (auto recognise) 4x6 keyboard interface is provided Up to 460 kbps
Keyboard interface	
UART0 interface with flow control	Full hardware flow control signals (+3.0V) are provided on 60 pins.
	2-Wire UART interface
UART1 interface without flow control	Up to 460 kbps
	Support standard SPI interface
LCD interface	

1.4 FUNCTIONAL DESCRIPTION

1.4.1 RF Functionalities

The RF part of this module converts RF signals to baseband for receiver chain and translates base band signals into RF frequency spectrum.

The operating frequencies are:

Rx (EGSM 850): 869 to 894MHz

Tx (EGSM 850): 824 to 849MHz

Rx (EGSM 900): 925 to 960MHz

Tx (EGSM 900): 880 to 915MHz

Rx (DCS 1800): 1805 to 1880MHz

Tx (DCS 1800): 1710 to 1785MHz

Rx (PCS 1900): 1930 to 1990MHz

Tx (PCS 1900): 1850 to 1910MHz

1.4.2 Baseband Functionalities

The baseband part of SM5210 is composed of a SPREADTRUM's SC6600D chip. This chipset is using 0.18 μ m mixed signal CMOS technology which allows massive integration as well as low power consumption. SC6600D provides single-chip solution to wireless Quad-band telephone handsets and data modems confirming to the EGSM 900, GSM 850, DCS 1800 and PCS 1900.

1.4.3 Memory Functionalities

The memory used in this module is a combination of 32 Megabit (2M x 16-Bit) CMOS 3.0V Volt-only, simultaneous operation flash memory and 4 Megabit (256K x 16-Bit) static RAM.

2 INTERFACE AND FUNCTIONALITIES

2.1 POWER SUPPLY

2.1.1 Power Supply Description

The power supply is one of the key issues in the designing GSM terminals. Due to the 577us radio burst emission in GSM every 4.615ms, power supply must be able to deliver high current peaks in a short time. During these peaks, ripples and drops on the supply voltage must not exceed a certain limit.

Table 1: VBAT Pin Description

	Pin number	V _{MIN}	V _{NOM}	V _{MAX}	Ripple max
VBAT (V)	1,3,5,7	3.3V (*)	3.6V	4.2V (**)	50mV _{PP} for freq<200KHz 2mV _{PP} for freq>200KHz

(*): This value has to be guaranteed during the burst (with 2.0A Peak in GSM or GPRS mode).

(**): max operation Voltage Stationary Wave Ratio.

When supplying the module with a battery, the total impedance (battery + protections + PCBS) should be less than 150mOhms.

Table 2: GND Pin Description

	Pin number	Description
GND	9,11,13,15,17 Shielding can install pin(*)	Ground

5.8 Servo motor

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system. The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller.

If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

A servomotor is closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is some signal, either analog or digital, representing the position commanded for the output shaft

CHAPTER 6

RESULT AND DISCUSSION

6.1 Result and Discussion

For getting better understanding of proposed model so we need some experimental setup. Who are authorised by admin firstly stands in front of camera and it will capture the image of person and stored in the admin database images. If the stored images is recognized with captured image the door will be lock/unlocked. When person image not recognized then system send capture image to the admin and wait for the OTP by admin. If admin allow with OTP to the system then system allow the person for lock/unlock and store the image

CHAPTER 7

CONCLUSION AND FUTURE WORK

Smart door locking/unlocking system will help in developing keyless door locking/unlocking and also remote door unlocking. This IOT based device will remove the need of manually locking/unlocking the door for the registered user. Also for an unknown user, this device will provide an extra layer of security for the residents of the house. An alert message will be immediately send to the registered user when unknown face is detected. This will help in user to decide whether or not to allow a particular person

CHAPTER 8

REFERENCE

1. A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007.
2. L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, pp. 407–415, IEEE, Fort Worth, Tex, USA, March 2003.
3. . L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, pp. 407–415, IEEE, Fort Worth, Tex, USA, March 2003.

4. A. Oka and L. Lampe, “Distributed target tracking using signal strength measurements by a wireless sensor network,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1006–1015, 2010.
5. P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF based user location and tracking system,” in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 775–784, IEEE, Tel Aviv, Israel, March 2000
6. Z. Dian and L. M. Ni, “Dynamic clustering for tracking multiple transceiver-free objects,” in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, pp. 1–8, Galveston, Tex, USA, March 2009. 7
7. Z. Dian and L. M. Ni, “Dynamic clustering for tracking multiple transceiver-free objects,” in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, pp. 1–8, Galveston, Tex, USA, March 2009. 7

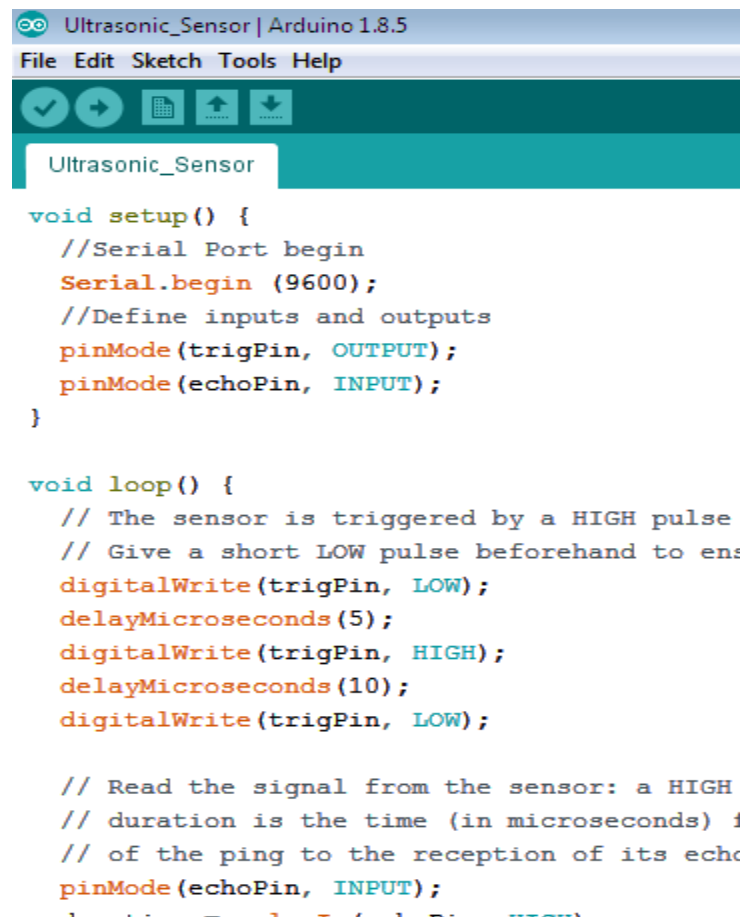
CHAPTER 8

SAMPLE CODE

The Mega 2560 board can be programmed with the Arduino Software (IDE). For details, see the reference and tutorials.

The ATmega2560 on the Mega 2560 comes preprogrammed with a boot loader 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 boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

A screenshot of the Arduino IDE interface. The title bar reads "Ultrasonic_Sensor | Arduino 1.8.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a grid, an upload arrow, and a download arrow. A tab labeled "Ultrasonic_Sensor" is active. The main text area contains the following C++ code:

```
void setup() {
  //Serial Port begin
  Serial.begin (9600);
  //Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  // The sensor is triggered by a HIGH pulse
  // Give a short LOW pulse beforehand to ens
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the signal from the sensor: a HIGH
  // duration is the time (in microseconds) t
  // of the ping to the reception of its echo
  pinMode(echoPin, INPUT);
  // ... = ... / ...
```

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The

ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

- **On Rev1 boards:** connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- **On Rev2 or later boards:** there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader). See this user-contributed tutorial for more information.

WARNINGS:

The Mega 2560 has a resettable poly fuse 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.

POWER:

The Mega 2560 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 become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **Vin.** The input voltage to the 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.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.
- **IOREF.** This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source

or enable voltage translators on the outputs for working with the 5V or 3.3V.