UNIVERSITY OF NAIROBI

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

FINAL YEAR PROJECT REPORT

PROJECT INDEX: PRJ 071

ULTRASONIC DISTANCE MEASUREMENT BASED ON 8051 MICROCONTROLLER

BY

JEREMY W MOMANYI INYEGA

REGISTRATION NUMBER: F17/1323/2018

SUPERVISOR: PROF E. MWANGI

EXAMINER: PROF G. KAMUCHA

SUBMITTED ON: 9TH OF JUNE 2023

This project report is submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Electrical and Electronic Engineering from the University of Nairobi

DECLARATION OF ORIGINALITY

FACULTY/ SCHOOL/ INSTITUTE: Engineering.

DEPARTMENT: Electrical and Information Engineering.

COURSE NAME: Bachelor of Science in Electrical and Electronic Engineering

TITLE OF NAME OF STUDENT: Jeremy W Momanyi Inyega.

REGISTRATION NUMBER: F17/1323/2018

COLLEGE: Architecture and Engineering

WORK: Ultrasonic Distance Measurement System

1. I am aware of what it means to plagiarize someone else's work and that the University policy does not support plagiarism.
2. I certify that this final-year project report is entirely original and has not been submitted to be considered for another exam, to be granted a degree, or to be published. I have referenced and cited other people's work where it was necessary to do so.
3. I wrote and finished this project correctly without consulting any outside professionals.
4. I will not permit anyone to pass off this work as their own.
5. I am aware that violating the university's anti-plagiarism policy will result in disciplinary action.

Signature…………….…………………………… Date……………………………………

CERTIFICATION

This report has been submitted to the Department of Electrical and Information Engineering, the University of Nairobi with my approval as the supervisor:

Supervisor: Prof. Elijah Mwangi

Signature…………….…………………………… Date……………………………………

DEDICATION

I would like to dedicate this project to my parents, who have stood by me all these years, to my supervisor who provided guidance keeping me on track, my classmates who were always ready to lend a hand and last but greatest, I dedicate this project to God who has sustained me though my life.

ACKNOWLEDGEMENT

I give thanks to God for enabling me to reach the end of this project and the end of this course in general.. He has given me the strength and willpower to keep on going when it got tough.

I am grateful to my supervisor, Professor Elijah Mwangi. He provided us direction and a insight on how best to complete the project.

I am grateful to my predecessors who have done and documented various approaches to this project. My research would be lacking without their contributions to the field of Ultrasonics. I have provided links to their references.

I want to express my gratitude to the professors and students at the University of Nairobi, Department of Electrical and electronic engineering who were always willing to assist me and give me new ideas on how best to approach my project.

ABSTRACT

This project aims to implement an ultrasonic distance measurement system based on the 8051 microcontroller with the objective of accurately measuring distances using ultrasonic waves and providing real-time feedback. The system's basic components include an ultrasonic sensor, an 8051 microcontroller, and a display unit. The ultrasonic sensor sends out an ultrasonic pulse (trigger pulse) and detects the echo pulse reflected from the target object. The sensor then outputs an electrical signal to the microcontroller. The microcontroller, 8051 based for this system, acts as the central control unit of the system. It communicates with the ultrasonic sensor, telling it to send out the trigger pulse and the sensor responds after a certain amount of time signaling receipt of the echo pulse. The microcontroller utilizes its onboard timers and interrupts to calculate the time taken for the ultrasonic waves to travel to the target object and back. The microcontroller uses this determined time and calculates the distance to the target object using known speed of sound in air. Having determined the distance, the microcontroller sends this value to a suitable display unit, such as an LCD screen, where the distance is displayed in a user-friendly format, allowing for easy interpretation and observation. The implemented system offers several advantages, including non-contact distance measurement, high accuracy, and real-time feedback. It has various applications, such as in robotics, automation, and security systems. In conclusion, this project successfully demonstrates an ultrasonic distance measurement system based on the 8051 microcontroller. It provides an effective and reliable solution for measuring distances using ultrasonic waves, offering a wide range of potential applications.

NOTATION

LCD – Liquid Crystal Display

IC – Integrated Circuit

SoC - System on a Chip

ISR – Interrupt Service Routine

IVT – Interrupt Vector Table

Rx – Receive serial communication

Tx – Transmit serial communication

TTL – Time To Live

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1. BACKGROUND

In recent years, advancements in microcontroller technology have allowed them to be used for innovative applications in various fields. One such application is ultrasonic distance measurement, which offers an accurate, non-contact method for determining distances. Ultrasonic distance measurement finds applications in various fields, including robotics, automation, security systems, and object detection. This project focuses on developing an ultrasonic distance measurement system based on the 8051 microcontroller.

1.2. PROBLEM STATEMENT

The accurate measurement of distances is essential in many applications. Traditional methods such as measuring tapes or rulers have limitations in terms of precision, efficiency, range or response time. Therefore, there is a need for a reliable and automated distance measurement system that can overcome these limitations. This project aims to develop an ultrasonic distance measurement system using the 8051 microcontroller to address this problem.

1.3. SIGNIFICANCE OF THE PROJECT

This project holds significance for several reasons. Firstly, it provides a practical solution for accurate distance measurement without physical contact, allowing for safer and more efficient measurements. Secondly, the project allows for the practical application of microcontroller programming and interfacing skills, which are highly relevant in the field of embedded systems. Finally, the project contributes to the body of knowledge in the field of distance measurement and provides a foundation for further research and development in this area.

1.4. SCOPE OF THE PROJECT

This project focuses on designing, developing, and evaluating an ultrasonic distance measurement system using the 8051 microcontroller. The hardware implementation includes selecting appropriate ultrasonic sensors, designing the necessary interfacing circuitry, and integrating them with the microcontroller. The software development encompasses coding algorithms for sensor control, time-of-flight measurement, and distance calculation. The project thus involves hardware design, software programming, and system integration. However, advanced features such as obstacle detection or complex data processing are beyond the scope of this project.

1.5. OBJECTIVES

1. Design a hardware system that incorporates the use of ultrasonic sensors, an 8051 microcontroller and a visual display module to measure distance.
2. Develop software algorithms enabling the microcontroller to control the ultrasonic sensor, measure the time of flight, and calculate the distance accurately.
3. Calibrate and test the system to ensure its accuracy and reliability in different operating conditions.
4. Evaluate the performance of the developed system and compare it with existing methods for distance measurement.

1.6. METHODOLOGY

The project will follow a systematic approach that involves the following steps:

1. Study and review the literature on ultrasonic distance measurement techniques, microcontroller programming, and interfacing.
2. Understand the principles of ultrasonic sensors and the 8051 microcontroller.
3. Select and procure the necessary components for the project, including the ultrasonic sensor, 8051 microcontroller development board, LCD display, and supporting circuitry.
4. Design and implement the hardware connections and circuitry.
5. Develop the software program for the microcontroller to interface with the ultrasonic sensor, calculate the distance, and display it on the LCD screen.
6. Test and validate the functionality of the developed system.

7. Evaluate the performance of the system by comparing the measured distances with a reference standard.
8. Make any necessary improvements or modifications based on the evaluation results.

1.7. ORGANIZATION OF THE THESIS

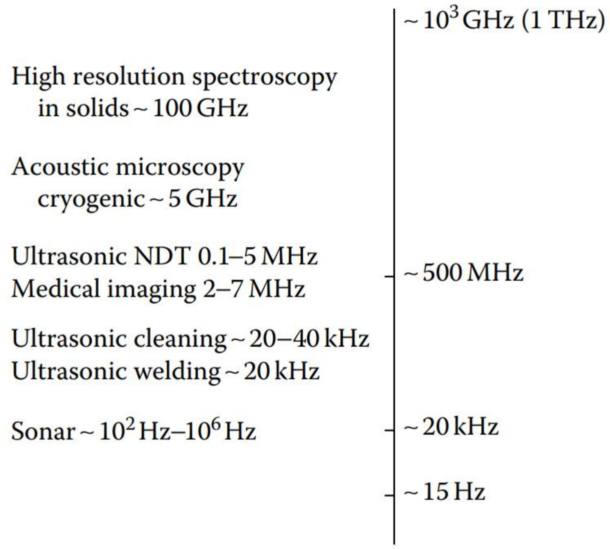
This rest of this thesis is organized as follows:

1. Chapter 2: Literature Review - This chapter presents a comprehensive review of relevant literature, including ultrasonic distance measurement techniques and microcontroller-based systems
2. Chapter 3: System Design - This chapter describes the design of the hardware system, including the selection of ultrasonic sensors, interfacing circuitry, and the specifications and interfaces of the 8051 microcontroller. It also details the software algorithms developed to control the ultrasonic sensors, measure time of flight, and calculate accurate distances using the microcontroller
3. Chapter 4: Hardware Implementation - This chapter presents the implementation details, including hardware connections and software programming.
4. Chapter 5: Results and Analysis – This chapter discusses the testing and evaluation of the developed system.
5. Chapter 6: Conclusion and Recommendations –This chapter concludes the document with a summary of the findings, limitations, and suggestions for future use.

CHAPTER TWO: LITERATURE REVIEW

2.1. ULTRASONIC MEASUREMENTS

Ultrasonic refers to soundwaves with a frequency greater than human hearing limit, that is, above 16 kHz. Thus, the field of ultrasonics is the study of the effects of propagation, matter interaction and application of sound waves at frequencies above human hearing [1].



Acoustic frequency scale and selected applications.

Figure 1: Frequencies used for select applications

For this project, we shall focus on distance measurement using ultrasonic waves. The basic principle followed involves an ultrasonic sensor sending out a pulse, which upon striking a sufficiently rigid object, reflects back to the sensor that can detect this echo signal.

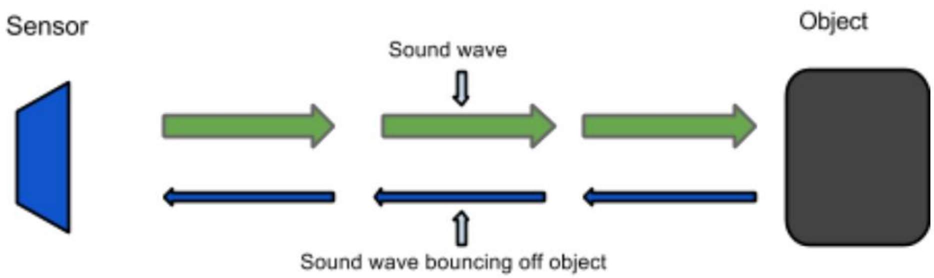


Figure 2: Basic Ultrasonic Distance Measurement Setup

2.1.1. Ultrasonic Sensors

These devices are capable of generating and detecting ultrasonic pulses.

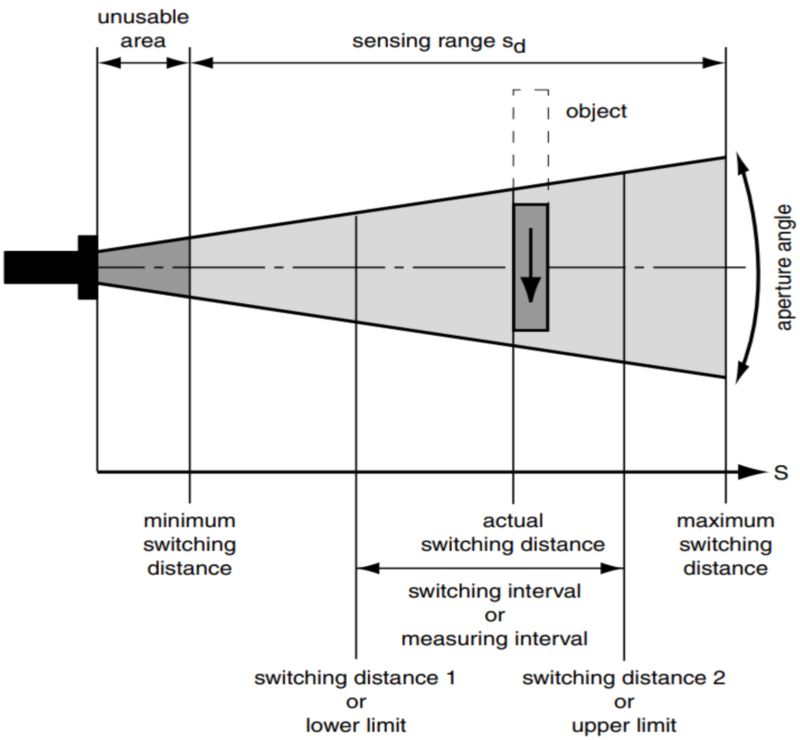


Figure 3: Usable Distance from ultrasonic transceiver

All ultrasonic sensors have an effective sensing range, S_d, a range bounded by the lowest and highest sensing distances. The sensor is able to detect objects within this range provided they remain within the sound cone. For best detection, it is best to be directly in front of the sensor as the sound pulse attenuates away from the center axis of the sensor.

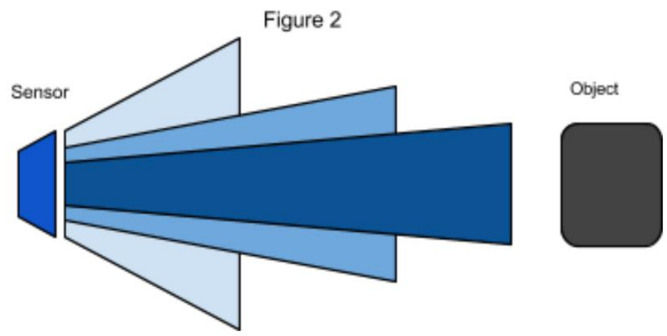


Figure 4: Effective Measurement Area

The diameter of the sound cone can be calculated using the equation:

$Diameter_{cone} = 2 * \tan\alpha * S$, where S is distance from sensor and α is sensor aperture angle

The lowest sensing distance bounds the unusable area of the sensor. This area exists due to switching time from Tx to Rx or delay by the Rx head waiting for crosstalk to fade.

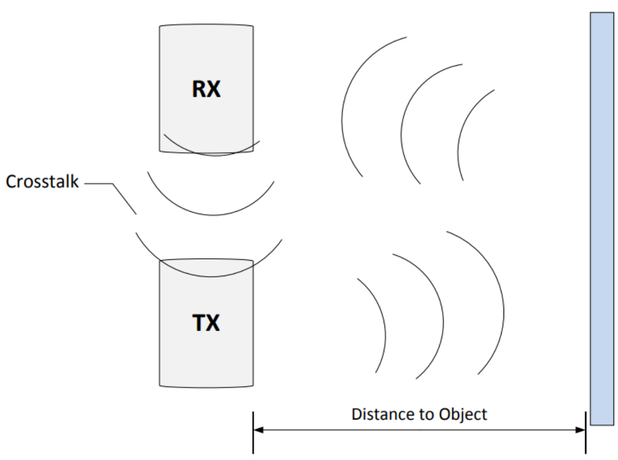


Figure 5: Crosstalk between Ultrasonic Transmitter and Receiver

This minimum distance is calculated using the equations [2]:

- $Total\ Receiver\ Wait\ Time = Burst\ Duration + \frac{Rx-Tx\ Distance}{Speed\ of\ sound}$
- $Minimum\ Detectable\ Distance = \frac{Total\ Receiver\ Wait\ Time * Speed\ of\ Sound}{2}$

In practice, this area should be less than 10% of S_d.

2.1.2. Possible transceiver configurations

1. Reflex Sensors
    - Uses a fixed reflector plate thus object detection is due to change in measured distance or lack of echo signal

Figure 6: Reflex sensor

2. Reflection Sensors
    - The object acts as the sound reflector thus sensing range is dependent on reflectance of the object (surface properties and angle of incidence)

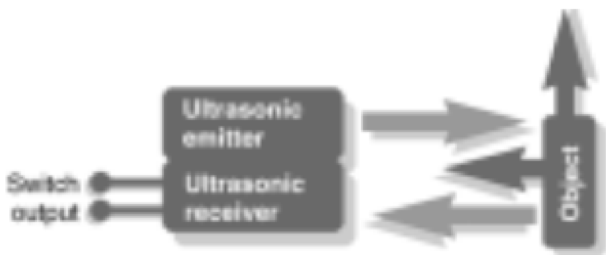


Figure 7: Reflection sensors

3. Twin head Reflection Sensors
    - Acts like a normal reflection sensor but has the ability to detect within a 3D space

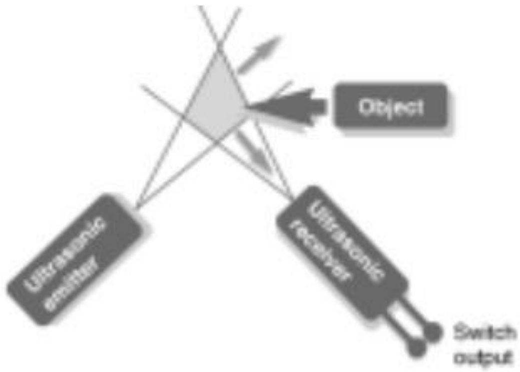


Figure 8: Twin-head reflection sensor

2.1.3. Advantages of Ultrasonic Distance Measurement

1. Fast measurement since it operates at the speed of sound allowing for rapid measurements to be executed
2. Unaffected by colour or transparency of objects as it only requires a sufficiently rigid body to reflect sound
3. Can be used in dark environments since it needs only sufficient air to operate
4. Low cost of ultrasonic modules allowing for cheaper implementation
5. Able to measure in dusty, dirty or humid environments – they may be able to enhance propagation of sound waves
6. Decent range – typically 3 – 4 m but some can reach over 15m

2.1.4. Limits of Ultrasonic Distance Measurement

1. Air temperature has the greatest impact on measurement as speed of sound is directly proportional to air temperature
2. Reflectivity - Soft materials absorb sound waves instead of reflect them back while objects at oblique angle may scatter the echo pulse instead of reflect it to sensor Rx
3. Humidity - at higher air temperatures, the speed of sound increases as humidity increases and can have a negative effect on accuracy [3]
4. Air Currents – strong winds may cause unstable measurements or even total signal loss

5. External noise – can generate false readings if at the same frequency as sensor leading to inaccurate measurements
6. Type of gas – different gas parameters from design may have an effect on accuracy
7. Echoes from distance object which can be picked up thus indicate a closer reading than expected
8. Cannot operate in vacuum since sound cannot travel in a vacuum

2.1.5. Ultrasonic Measurement Based On 8051

There exists a range of ultrasonic sensors available that can be interfaced with the 8051 microcontroller. These include:

- HC-SR04
- HY-SRF05
- Maxbotix LV-MaxSonar-EZ series
- Ping))) Ultrasonic Distance Sensor
- SRF04
- Parallax PING)))® Ultrasonic Sensor
- DYP-ME007Y
- JSN-SR04T
- MB7060 XL-MaxSonar-WR1
- Grove - Ultrasonic Ranger
- LV-Ultrasonic Distance Sensor
- Ultrasonic sensor module USSM1.0 PLUS-FS and many more

This project will focus on using the HC-SR04 ultrasonic module due to its low cost and easy availability.

Figure 9: HC-SR04 Ultrasonic module

It has the following specifications [2]:

- Working Voltage: DC 5 V
- Working Current: 15mA
- Working Frequency: 40kHz
- Max Range: 4m
- Min Range: 2cm
- Measuring Angle: 15°
- Trigger Input Signal: 10uS TTL pulse
- Echo Output Signal: Input TTL lever signal and the range in proportion
- Dimension: 45*20*15mm

The module has the following timing diagram:

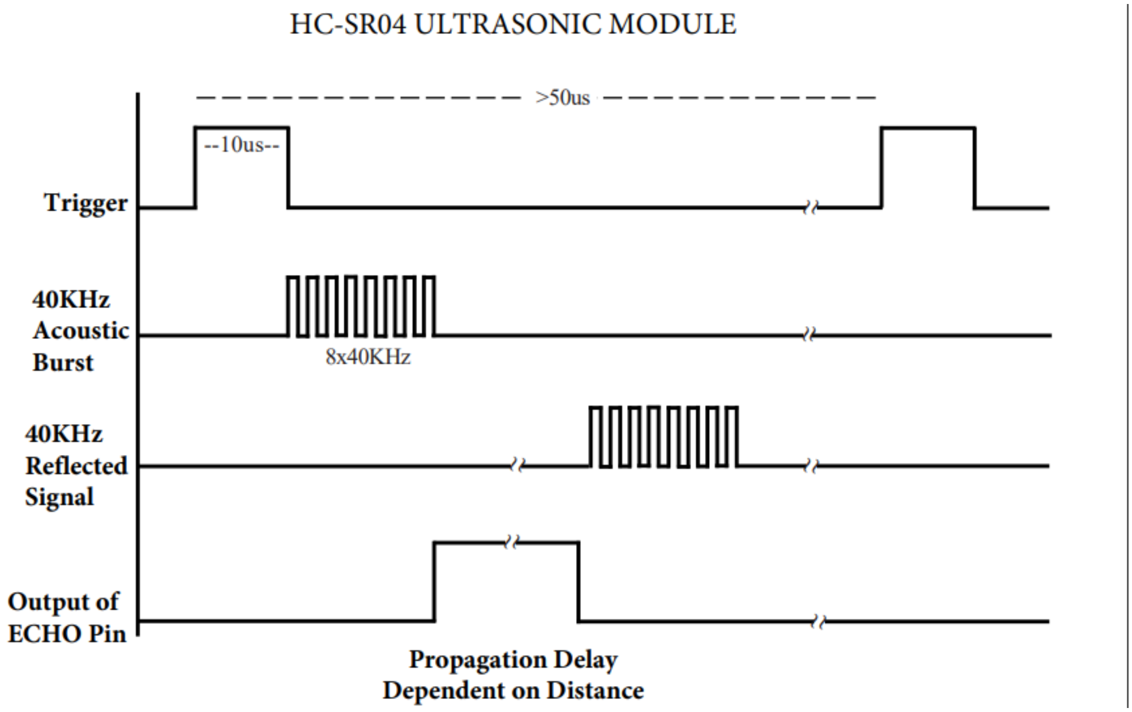


Figure 10: HC-SR04 Timing Diagram

Note that measurements can only be taken at a maximum of every 50 µs.

In order to take a measurement, we follow the following steps:

1. Transmit a trigger pulse of at least 10 µs to the HC-SR04 Trig Pin.
2. When the correct length trigger pulse is applied, the HC-SR04 automatically sends out a sonic burst consisting of eight 40 kHz sound waves and waits for the echo burst.
3. Start Timer of 8051 when Echo pin goes high and again wait for the falling edge on the Echo pin.
4. When the Echo pin goes low, the microcontroller reads the count of the timer. The timer count is thus used to calculate distance to the object as in the next section.

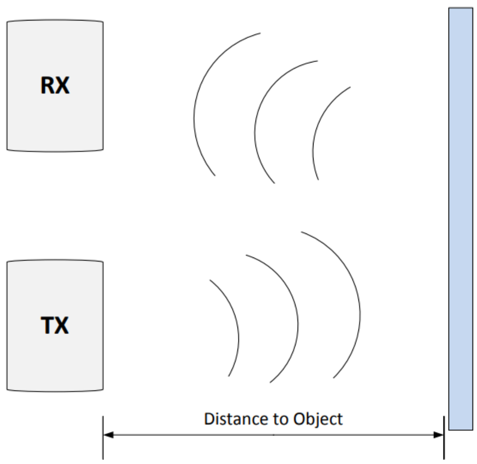


Figure 11: Simplified Ultrasonic distance transceiver

2.1.6. Distance Calculation

For normal operations, the distance travelled by the sound wave is calculated as:

$$Distance = \frac{Sound\ Velocity * Time\ of\ Flight}{2}$$

The sound velocity is approximated using the equation:

$$Sound\ Velocity = 331.4 + (0.6 * Temperature\ in\ °C) M/s$$

In the 8051 microcontroller, the time of flight is calculated from the inbuilt timers as:

$$Time\ of\ Flight = Timer\ count * Clock\ period$$

Thus the distance calculation for the system is:

$$Distance = \frac{Sound\ Velocity * Timer\ count * Clock\ period}{2}$$

2.2. THE 8051 MICROCONTROLLER

A microcontroller is a small IC that contains a processor core, memory, and programmable input/output peripherals acting as a computer on a single chip. Microcontrollers are widely used in embedded systems, which are dedicated computing systems designed for specific tasks or applications.

Microcontrollers come in various architectures, such as 8-bit, 16-bit, and 32-bit, each offering different capabilities and performance levels. They can be programmed to perform a wide range of functions, making them a versatile choice for numerous applications that require control and automation.

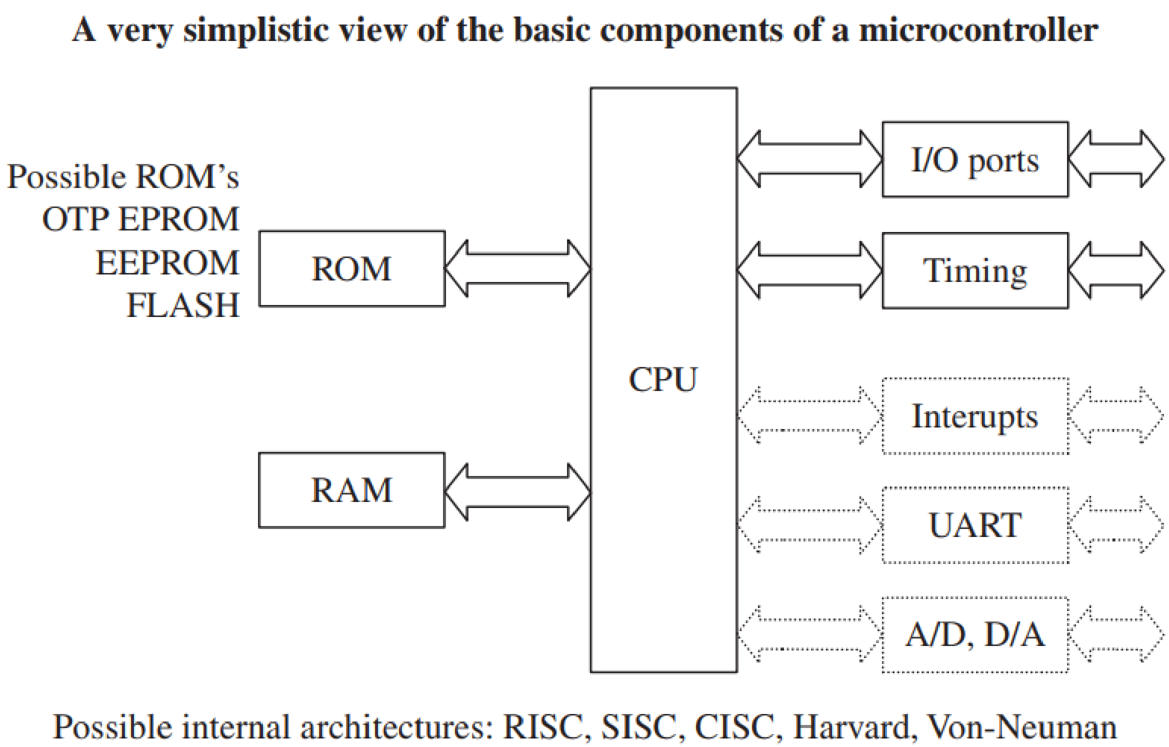


Figure 12: Simplified microcontroller structure

2.2.1. Criteria for choosing a microcontroller

1. Efficiency in terms of speed, packaging, power, RAM and Rom, I/O pins, upgradability and cost per unit
2. Integration – Ease of developing products for and around it
3. Availability

2.2.2. 8051 Family

Among these microcontrollers, there exists the 8051 family of microcontrollers.

Introduced in 1981 by Intel, the 8051 launched with 128B RAM, 4KB on-chip ROM, two timers, one serial port and four 8-bit ports all on a single chip.

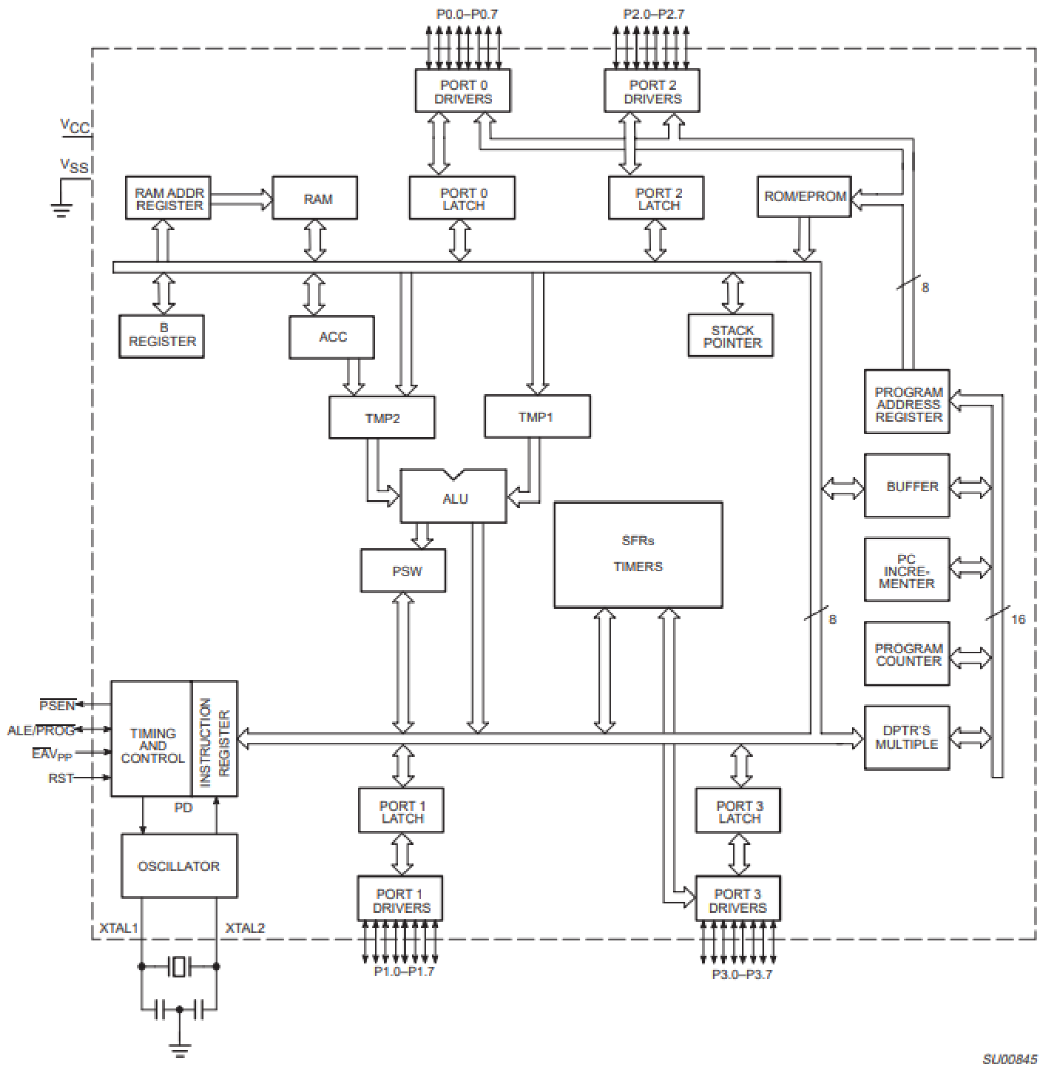


Figure 13: 8051 Microcontroller Architecture

While the 8051 is the most popular, 8051 microcontrollers are commonly found under different names depending on variations in the manufacturing process and manufacturer. Such name variants include:

- 8031 – ROM-less 8051
- 8052- improved 8051 with double RAM and ROM
- 8751 – UV –EPROM 8051
- 80C51 – Uses CMOS technology
- AT89C51 – 80C51 from ATMEL
- DS5000-8-12 – Made by Dallas Semiconductors

2.2.3. 8051 pinout

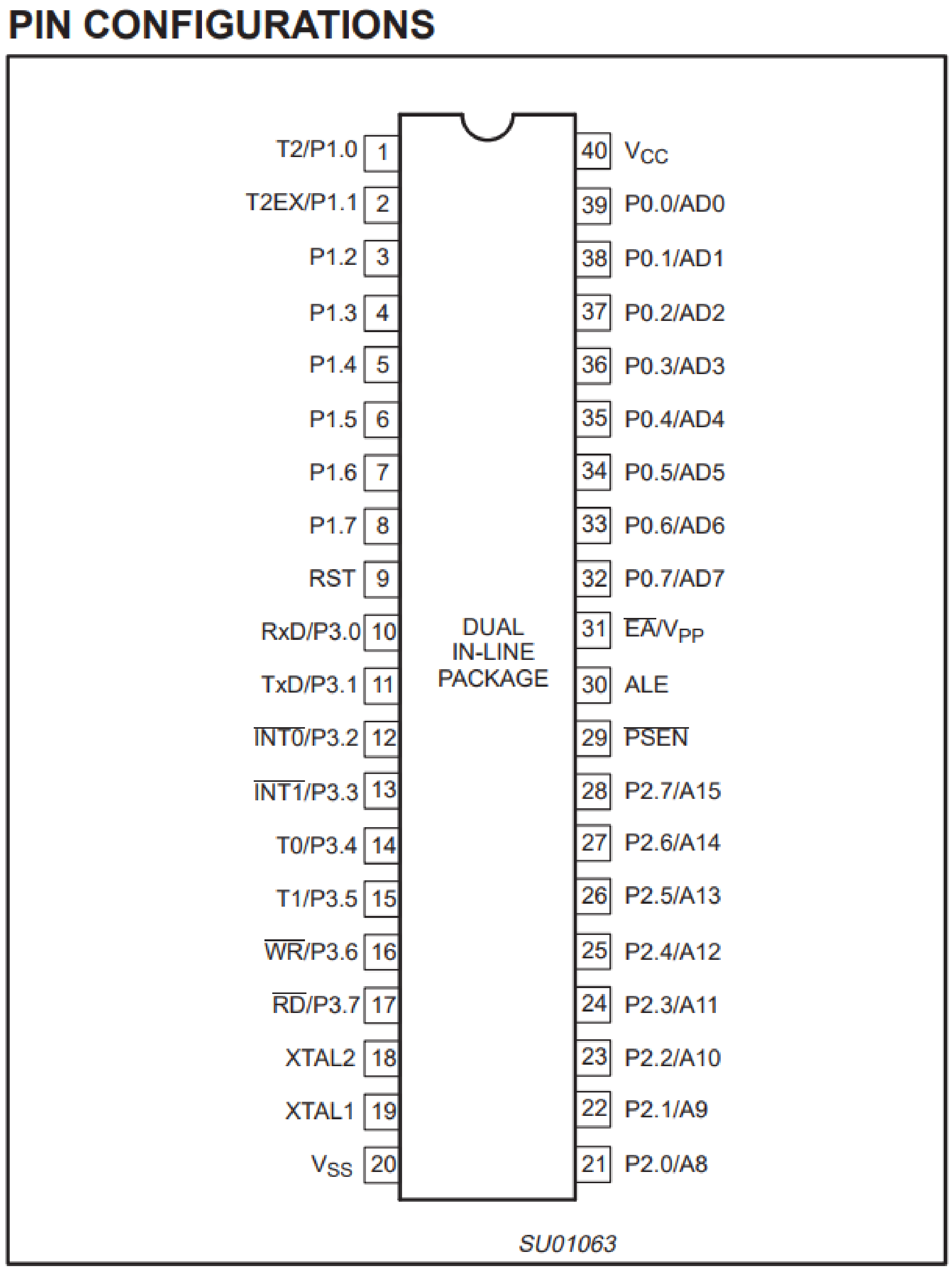


Figure 14: 8051 Pinout Diagram

The pins are used as follows:

1. Pin 40 (V_{cc})
    - Supplied with a high voltage, usually 5V
2. Pin 20 (V_{ss}/ Ground)
    - Connected to ground
3. Pin 9 (RST)
    - Rest button- clears program counter causing the program to start from the beginning
4. Pin 18 and 19 (XTAL2 and XTAL1)
    - Allow connection to an external oscillator used to set the desired chip frequency
5. Pin 31(EA/V_{pp})
    - Connected to V_{cc} to show that the ROM is loaded and code internally stored
6. Pin 30(PSEN)
    - Used in 8031
7. Pin 29(ALE)
    - Can be used for external clocking
8. Port 0 (pin 32-39)
    - Each pin is connected to V_{cc} via 10kΩ
    - Used as an I/O port
9. Port 1 (pin 1-8) and 2 (pin 21-28)
    - Work the same as Port 0 but without the need for pull up resistors.
10. Port 3 (pin10 – 17)
    - Works same as Port 1 and 2
    - Usually, used for interrupts

2.2.4. Machine cycles and Delay generation

Within a typical microcontroller system, there exists a need for delay generation and to generate a delay, an understanding of the machine cycle is necessary.

A machine cycle corresponds to a single increment in the program counter of the IC. It lasts 12 crystal oscillator periods hence:

$$frequency_{crystal} = 12 * \ frequency_{machine\ cycle}$$

The time taken by each instruction is thus:

$$Period_{machine\ cycle} = \ {}^{1}/{frequency_{machine\ cycle}} = \ {}^{12}/{frequency_{crystal}}$$

The time to execute a set of instructions can now be determined to be:

$$Execution\ time = Period_{machine\ cycle} * Number\ of\ Instructions$$

A time delay is calculated as:

$$Time_{delay} = \ Period_{machine\ cycle} * \ (Delay\ Cycles * Cycles\ per\ instruction + Initialization\ cycles + Return\ cycle)$$

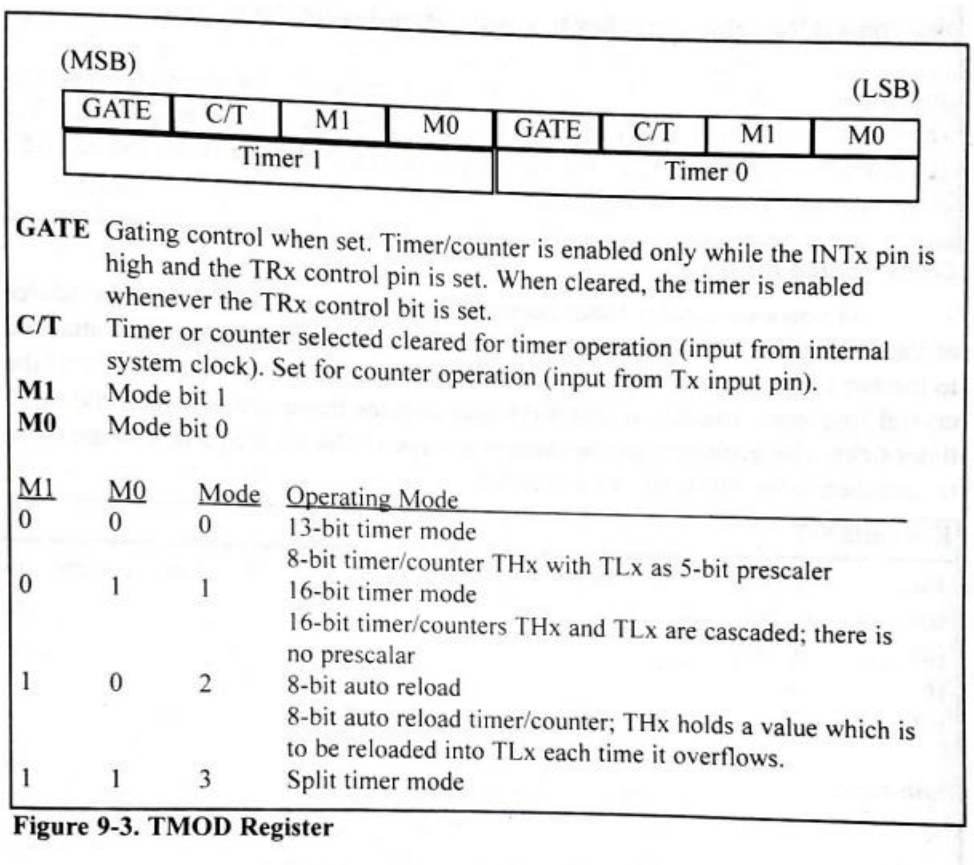
Usually accomplished using a loop such as:

```
D: MOV R3, # 100D

HERE: DJNZ R3, HERE ; two machine cycle instruction

RET
```

Where the delay is:

$$Time_{delay} = \ Period_{machine\ cycle} * \ (100 * 2 + 1 + 1)$$

2.2.5. Timers



(MSB) (LSB)

GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			

GATE Gating control when set. Timer/counter is enabled only while the INTx pin is high and the TRx control pin is set. When cleared, the timer is enabled whenever the TRx control bit is set.

C/T Timer or counter selected cleared for timer operation (input from internal system clock). Set for counter operation (input from Tx input pin).

M1 Mode bit 1

M0 Mode bit 0

M1	M0	Mode	Operating Mode
0	0	0	13-bit timer mode 8-bit timer/counter THx with TLx as 5-bit prescaler
0	1	1	16-bit timer mode 16-bit timer/counters THx and TLx are cascaded; there is no prescalar
1	0	2	8-bit auto reload 8-bit auto reload timer/counter; THx holds a value which is to be reloaded into TLx each time it overflows.
1	1	3	Split timer mode

Figure 9-3. TMOD Register

Figure 15: TMOD Register for timer setting

Note that timer frequency is 1/12 oscillator frequency.

To control the timers we use the TCON register:

1. Timer 0
   a. Overflow in detected by a 1 in TCON.5
   b. Timer is started by a 1 in TCON.4
2. Timer 1
   a. Overflow in detected by a 1 in TCON.7
   b. Timer is started by a 1 in TCON.6

D7							D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when timer/counter 1 overflows. Cleared by hardware as the processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn timer/counter 1 on/off.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when timer/counter 0 overflows. Cleared by hardware as the processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn timer/counter 0 on/off.
IE1	TCON.3	External interrupt 1 edge flag. Set by CPU when the external interrupt edge (H-to-L transition) is detected. Cleared by CPU when the interrupt is processed. *Note:* This flag does not latch low-level triggered interrupts.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low-level triggered external interrupt.
IE0	TCON.1	External interrupt 0 edge flag. Set by CPU when external interrupt (H-to-L transition) edge detected. Cleared by CPU when interrupt is processed. *Note:* This flag does not latch low-level triggered interrupts.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low-level triggered external interrupt.

Figure 11-6. TCON (Timer/Counter) Register (Bit-addressable)

Figure 16: TCON Register for timer control

2.2.6. Interrupts

An interrupt is an internal or external event that breaks the normal operation of a microcontroller indicating a device requires the processors attention.

In order to manage several devices, the microcontroller utilizes two methods:

1. Interrupt method
    - The device under control sends an interrupt pulse to the microcontroller. The microcontroller detects this signal, suspends normal operation to execute an ISR after which the microcontroller resumes normal operation.
2. Polling method
    - The microcontroller periodically checks in on the connected device to determine whether it needs attention or not

The microcontroller executes the following steps to handle an interrupt:

1. Finish current instruction then save the location of the next instruction to stack
2. Save all other interrupt statuses internally
3. Jump to the interrupt vector table holding the ISR address
4. Execute the ISR
5. Return to program by reading saved location of next instruction from stack (saved in step 1)

The 8051 microcontroller has the following interrupts:

1. Reset
    a. Tied to pin 9
    b. IVT located at 0000H
2. Timer 0
    a. Tied to timer 0 overflow
    b. IVT located at 000BH
3. Timer 1
    a. Tied to timer 1 overflow
    b. IVT located at 001BH
4. INT0
    a. External interrupt 0 tied to pin 12 (Port 3.2)
    b. IVT located at 0003H

5. INT1
    a. External interrupt 0 tied to pin 13 (Port 3.3)
    b. IVT located at 0013H
6. Serial communication (serves for Tx and Rx)
    a. IVT located at 0023H

These interrupts are enabled or disabled by the Interrupt Enable (IE) register:

D7 D0

EA	--	ET2	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
--	IE.6	Not implemented, reserved for future use. *
ET2	IE.5	Enables or disables timer 2 overflow or capture interrupt (8952).
ES	IE.4	Enables or disables the serial port interrupt.
ET1	IE.3	Enables or disables timer 1 overflow interrupt.
EX1	IE.2	Enables or disables external interrupt 1.
ET0	IE.1	Enables or disables timer 0 overflow interrupt.
EX0	IE.0	Enables or disables external interrupt 0.

* User software should not write 1s to reserved bits. These bits may be used in future Flash microcontrollers to invoke new features.

Figure 17: Interrupt enable Register

When an interrupt is detected, the interrupt flag register is updated, with each interrupt having its own flag:

Table 1: Interrupt flags

Interrupt	**Flag**	**SFR Register Bit**
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 0	TF0	TCON.5
Timer1	TF1	TCON.7
Serial Port	T1	SCON.1
Timer 2	TF2	TCON.7(AT89C52)
Timer 2	EXF2	TCON.6(AT89C52)

By default, these interrupts have the following priority:

1. External Interrupt 0 (Highest)
2. Timer 0
3. External Interrupt 1
4. Timer 1
5. Serial Communication (Lowest)

Due to the nature of priority levels, it is possible to program an interrupt within an interrupt.

2.2.7. Chip Selection

The chip chosen for this project was the STC89C52RC. This chip was chosen mainly due to its availability and cost compared to other chips like the AT89C51/2.

Figure 18: STC89C52RC Chip

STC89C52RC features [3]:

- Enhanced 80C51 Central Processing Unit ,6T or 12T per machine cycle
- Operation voltage range: 5.5V~3.3V
- Operation frequency range: 0- 48MHz@12T, or 0-24MHz@6T
- On-chip 8K FLASH program memory with flexible ISP/IAP capability
- On-chip 512 byte RAM
- Capable of addressing up to 64K byte of external RAM or external memory
- Dual Data Pointer (DPTR) to speed up data movement
- Three 16-bit timer/counter, Timer 2 is an up/down counter with programmable clock output on P1.0
- 8 vector-address, 4 level priority interrupt capability
- One enhanced UART with hardware address-recognition, frame-error detection function, and with self-baud-rate generator.
- One 15 bits Watch-Dog-Timer with 8-bit pre-scaler (one-time-enabled)
- integrate MAX810 - specialized reset circuit

- Three power management modes: idle mode and power-down mode
- Low EMI: inhibit ALE emission
- Power down mode can be woken-up by INT0/P3.2 pin, INT1/P3.3 pin, T0/P3.4, T1/P3.5, RXD/P3.0 pin
- Four 8-bit programmable bi-directional I/O ports
- Operating temperature: -40 ~ +85℃ (industrial) / 0~75℃ (commercial)
- package type : PDIP-40

2.2.8. Chip Programming

The hex file is flashed to the STC89C52RC chip using a circuit similar to the one below with U1 being a USB to Serial converter:

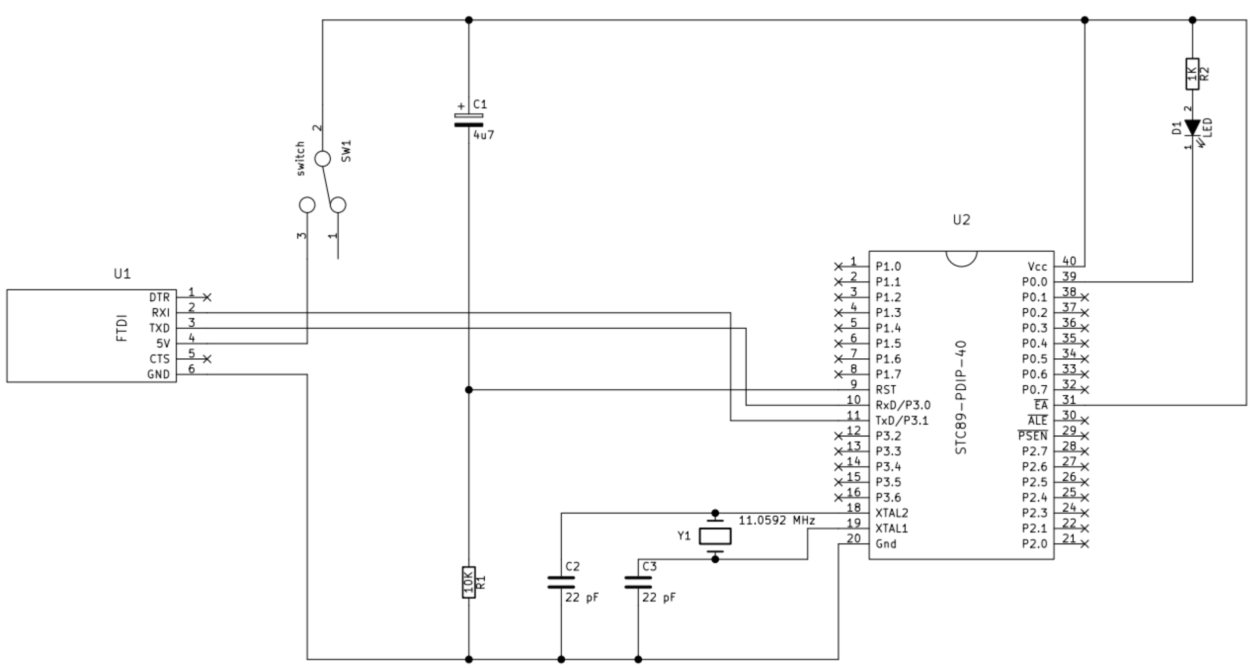


Figure 19: STC89C52RC Programming Circuit Diagram

2.3. LCD DISPLAY

A commonly used display module is the character LCD.

Figure 20: Character LCD displaying text

For this project, we shall restrict ourselves to the 16 by 2 LCD, though once learned the concepts here can be applied to any size character LCD with little to no modifications.

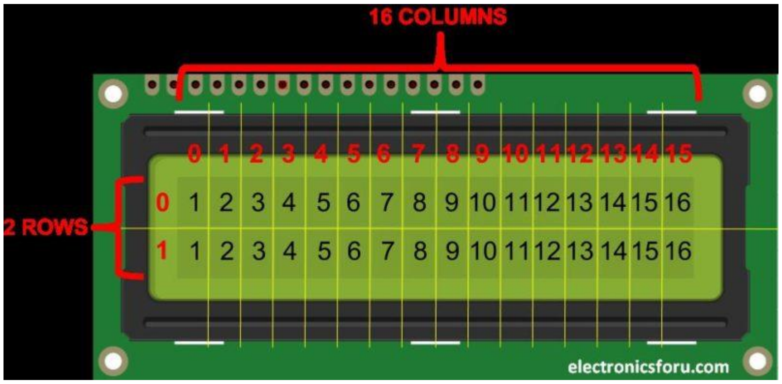


Figure 21: 16 x 2 LCD layout

The LCD has the following pinout:

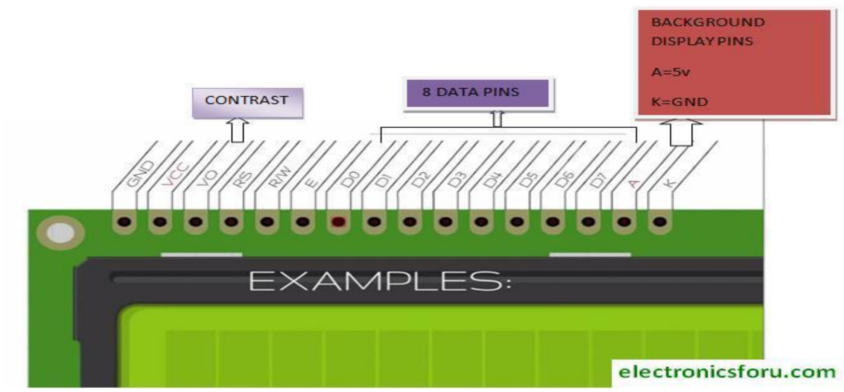


Figure 22: Character LCD Pinout

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Figure 23: LCD Pinout Description

To send data to the LCD, R/W low and E high while RS is held low for commands and high for data. Data is sent in form of ASCII digits while commands are 8- bit binary codes, some of which are outlined below:

Table 12-2: LCD Command Codes

Code (Hex)	**Command to LCD Instruction Register**
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 matrix

Note: This table is extracted from Table 12-4.

Figure 24: Common LCD Commands

CHAPTER 3: SYSTEM DESIGN

3.1. HARDWARE DESIGN

The following block diagram was developed as a guideline to build the system:

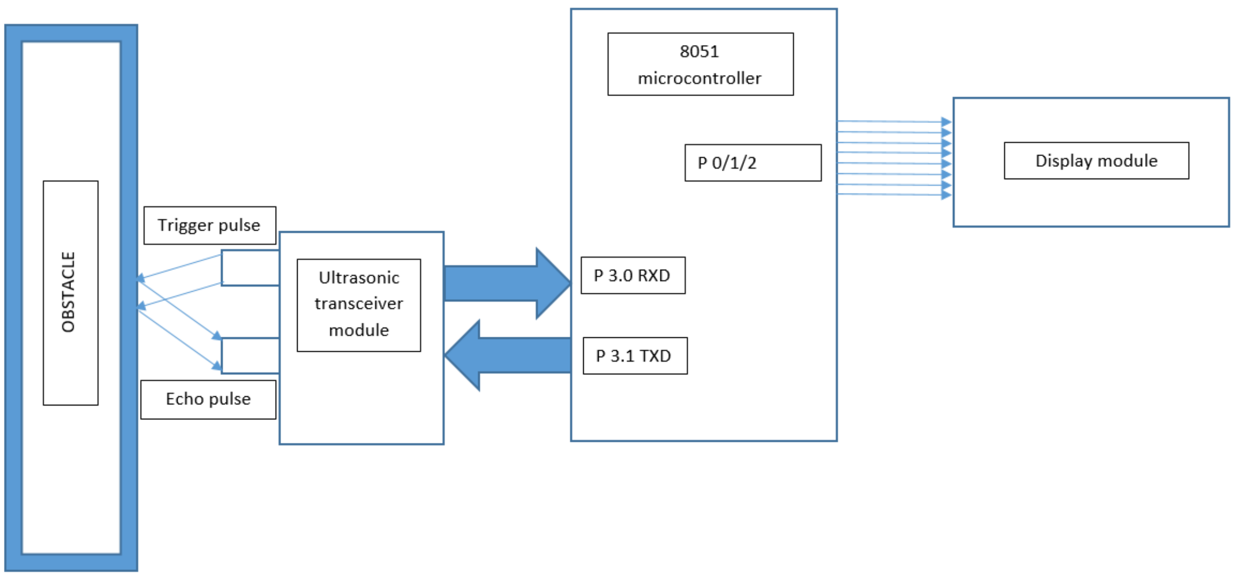


Figure 25: Ultrasonic Distance Measurement System Block Diagram

Based on information from their datasheets the following circuit diagram was developed to implement the system:

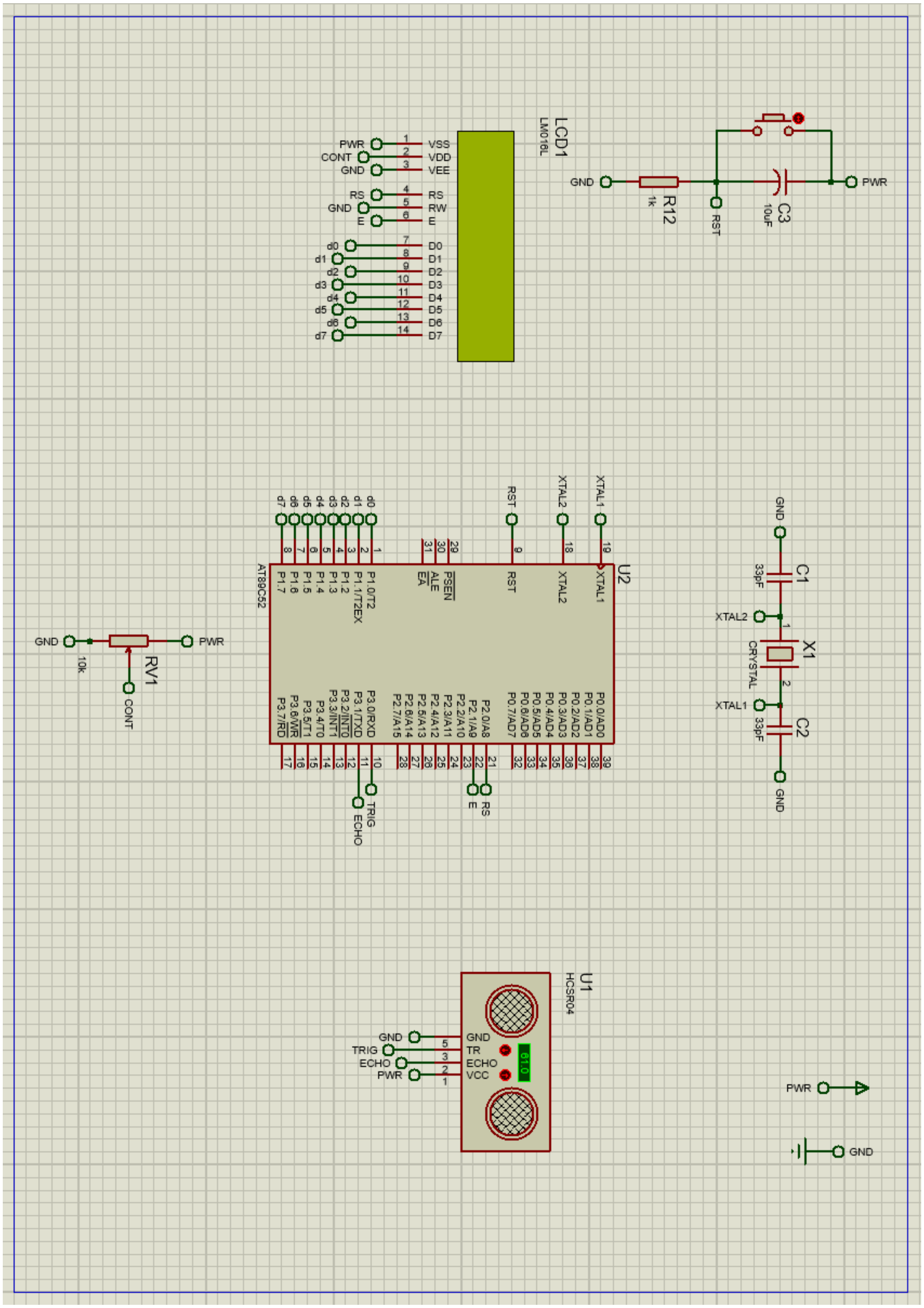


Figure 26: Ultrasonic Distance Measurement System Circuit Diagram

3.2. SOFTWARE DESIGN

3.2.1. General process flowchart

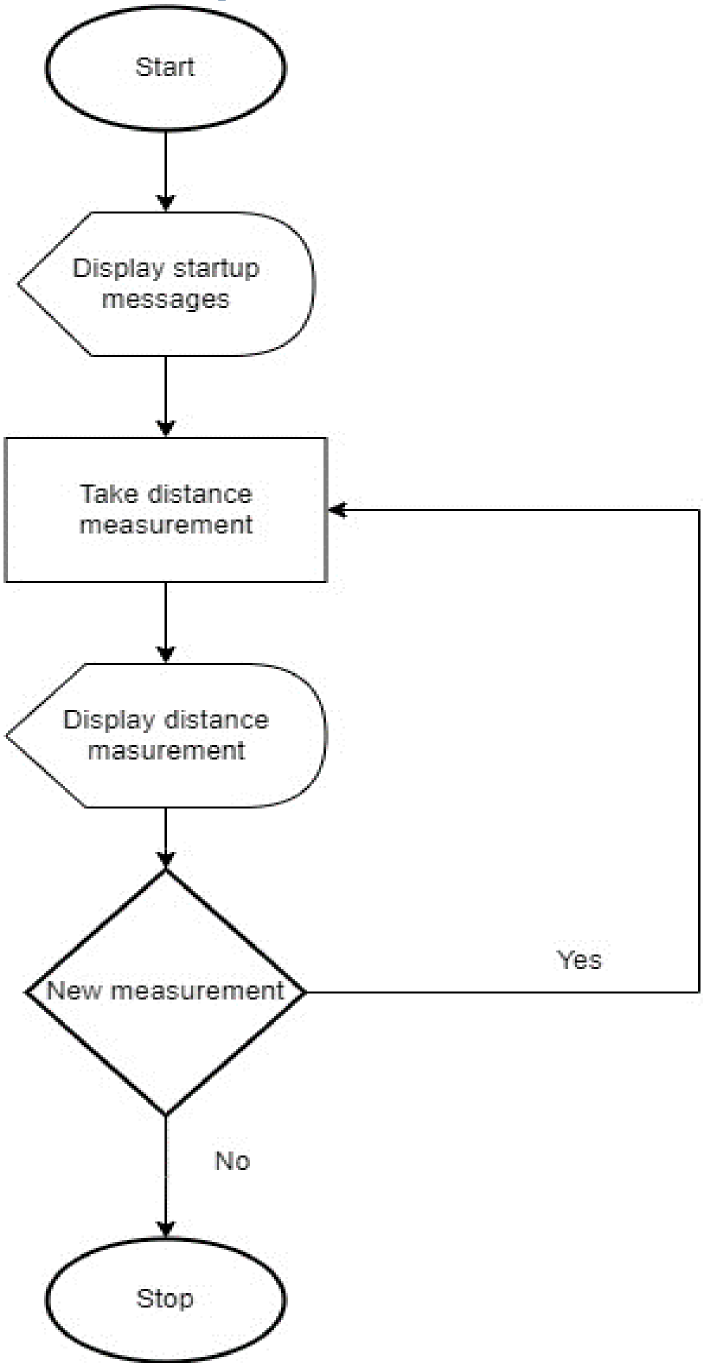


Figure 27: General process flowchart

3.2.2. Distance measurement flowchart

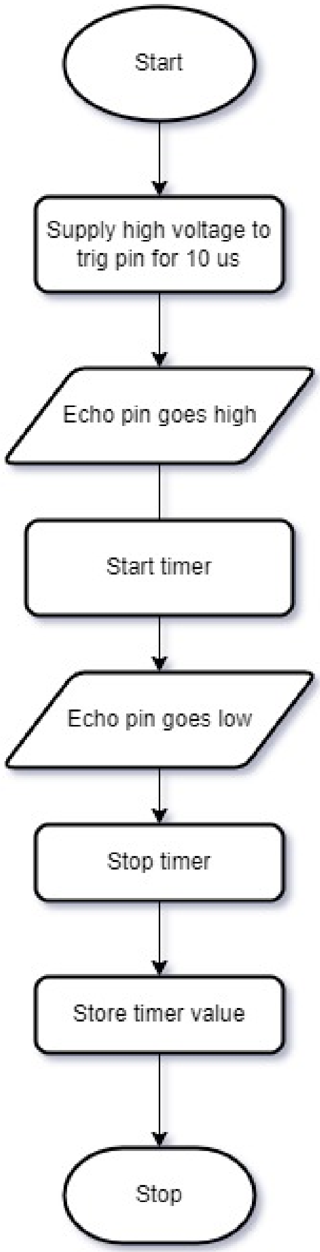


Figure 28: Distance measurement flowchar

3.2.3. Distance display flowchart

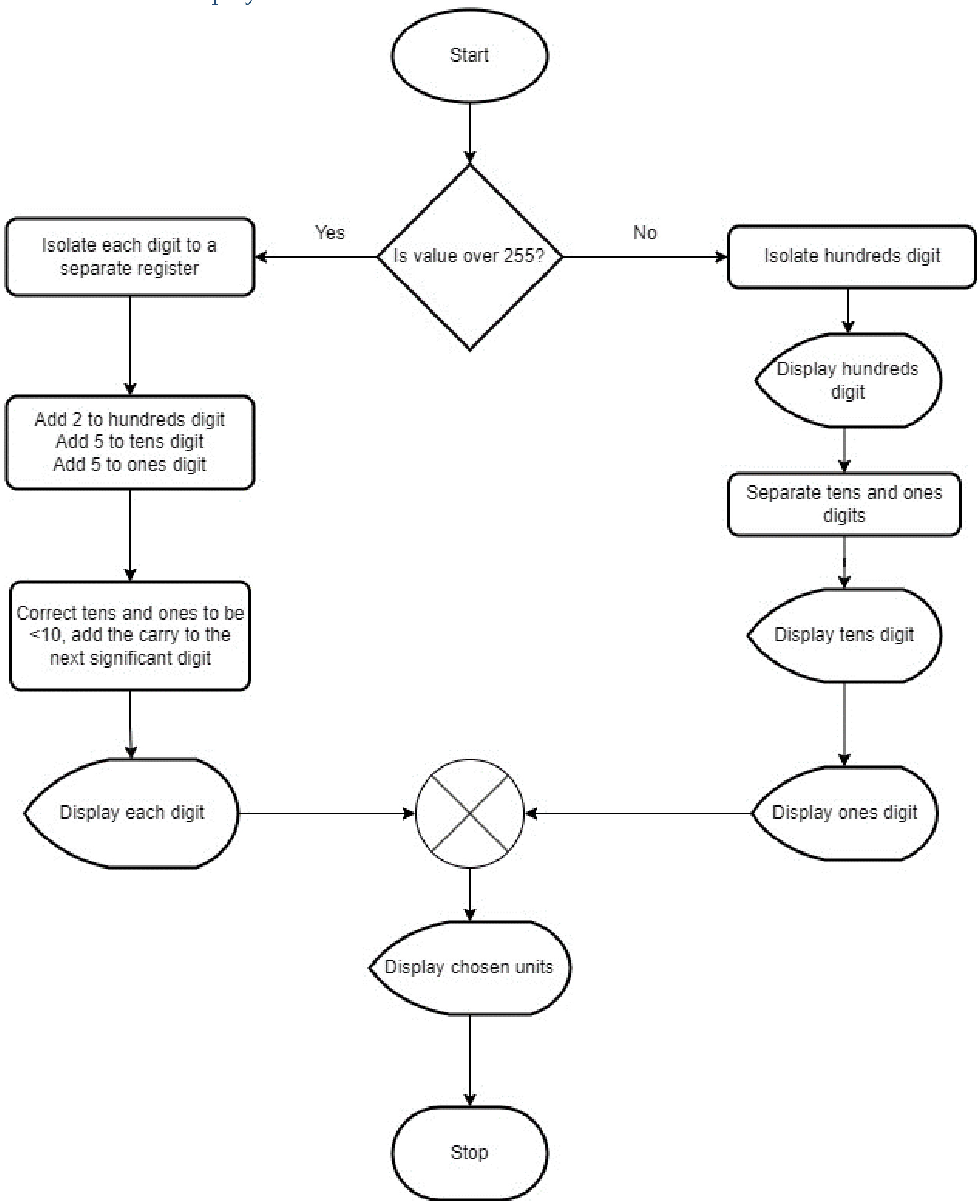


Figure 29: Distance display flowchart

CHAPTER 4: HARDWARE IMPLEMENTATION

4.1. UPLOADING CODE

The chip is connected to a USB to TTL/Serial module as shown below:

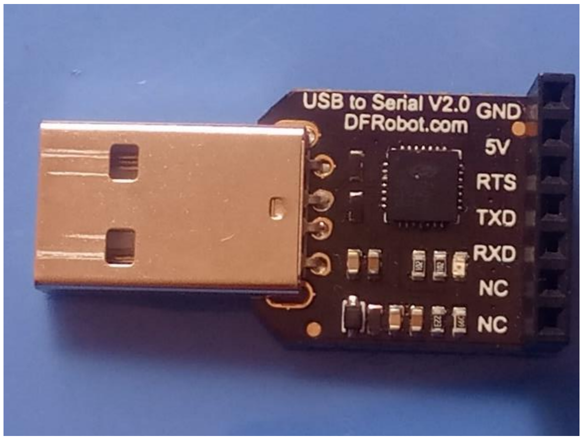


Figure 30: USB to serial converter

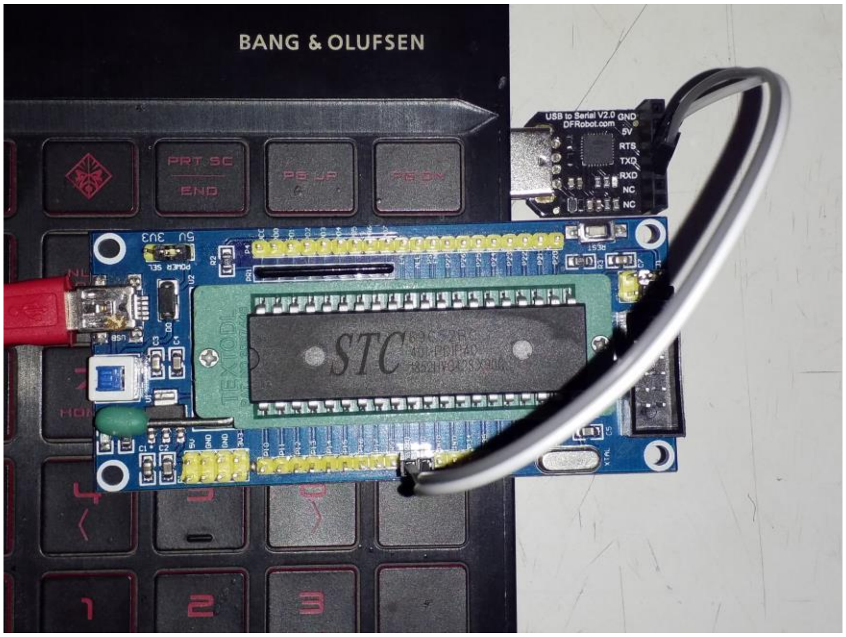


Figure 31: STC89C52RC Progamming circuit when plugged in

Once the USB to TTL is plugged into the computer USB port with the necessary software, the chip is programmed as follows:

1. Load Hex file to programmer software

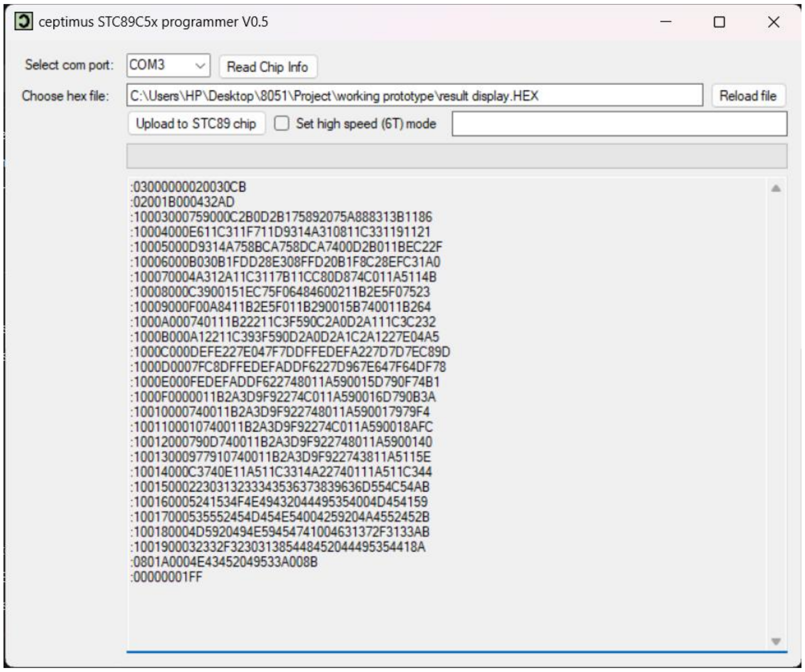


Figure 32: STC89C5x programmer loaded with hex file

2. Click Program then apply power

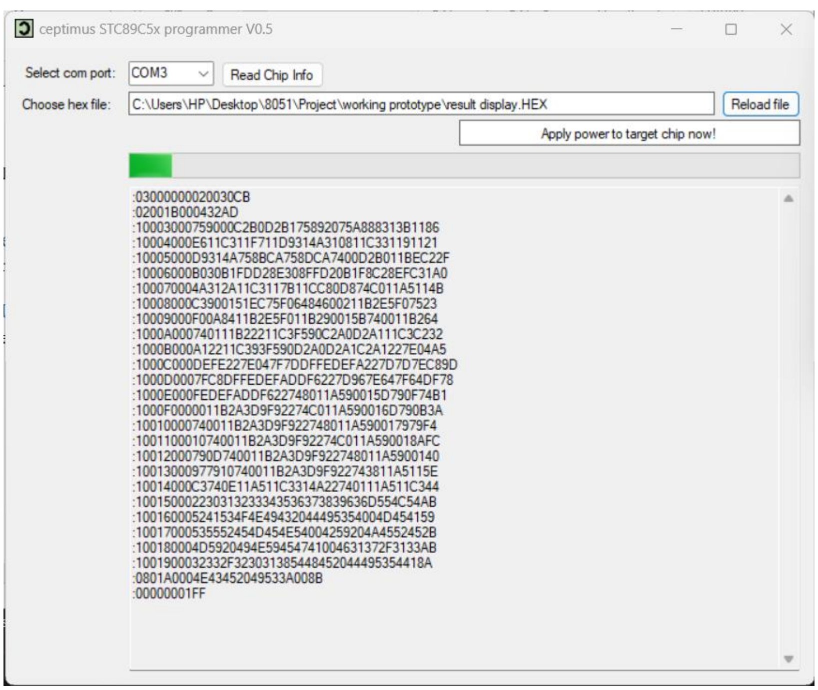


Figure 33: STC89C5x programmer prompt user to apply power

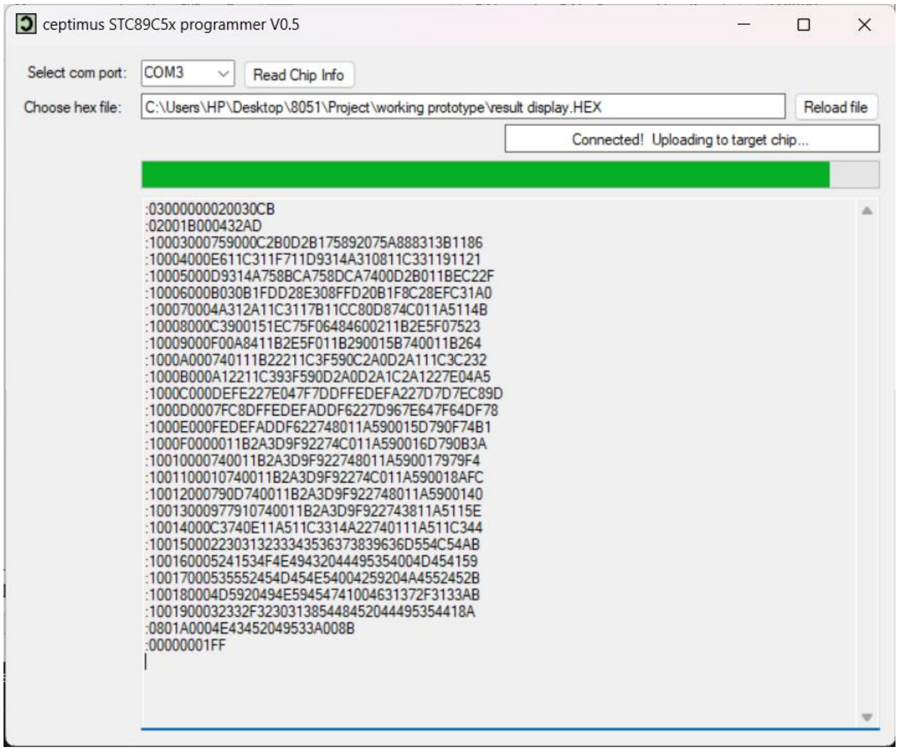


Figure 34: STC89C5x programmer successfully connected to target

3. Chip programmed

- It also obtains information about the chip at this stage

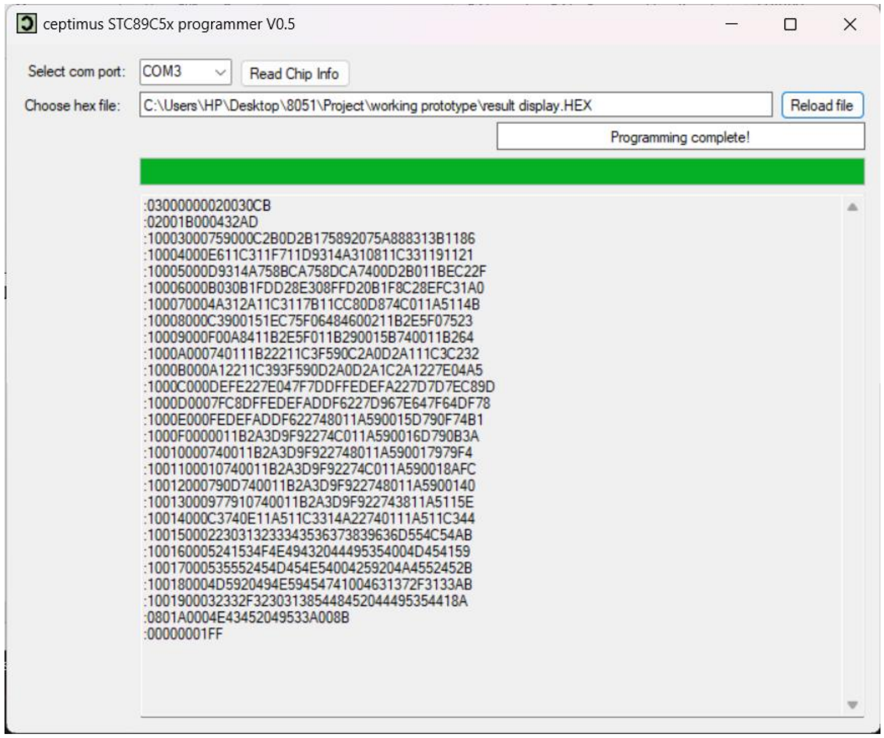


Figure 35: STC89C5x programmer successfully programmed chip

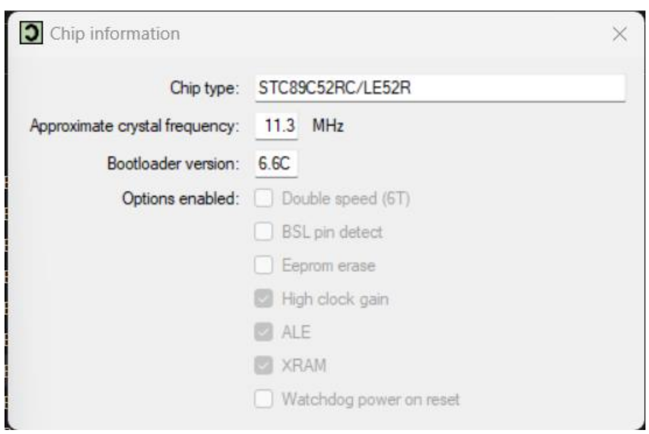


Figure 36: Chip information obtained by STC89C5x programmer

4.1.1. Software used

STC89C5xProgrammer

4.2. SIMULATIONS

Once uploaded to the Proteus 8 Professional circuit simulator the code displays the following when run:

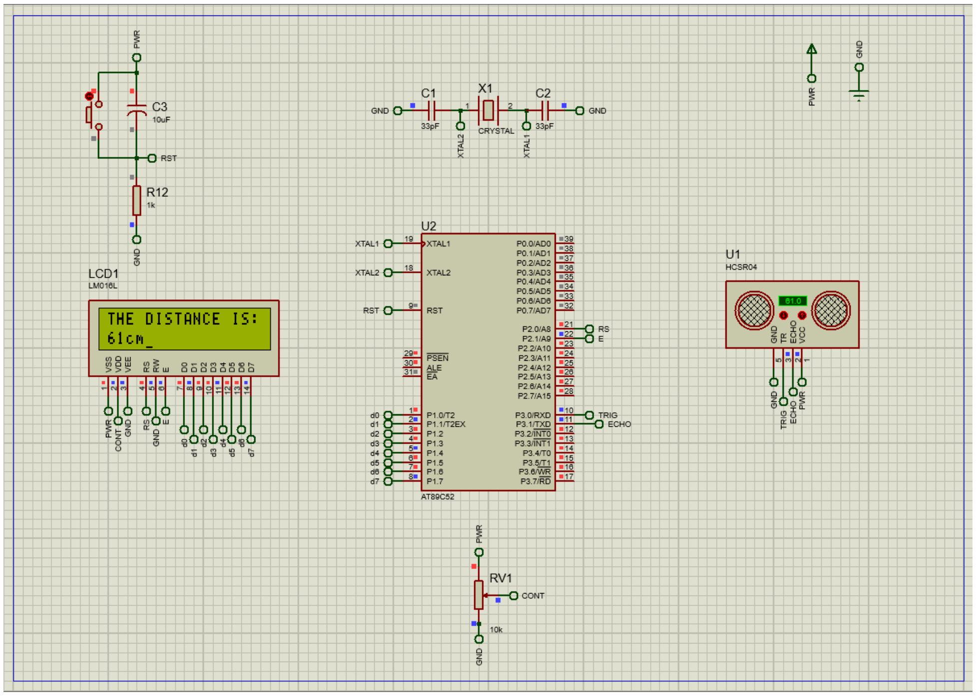


Figure 37: Simulation of Ultrasonic Distance Measurement system

The simulated sensor has been set to 61 cm that is picked up by the simulated chip thus indicating that the code is working.

4.3. PHYSICAL CIRCUIT

4.3.1. List of Components

1. STC89C52RC microcontroller
2. HCSR04 ultrasonic sensor
3. 16x2 LCD Display
4. 10 kΩ Potentiometer
5. Connecting wires
6. Breadboard
7. Power Supply
8. STC89C52 microcontroller seat (optional)

4.3.2. Circuit Images

Once programmed, the chip is connected to the physical realization of the circuit as shown below:

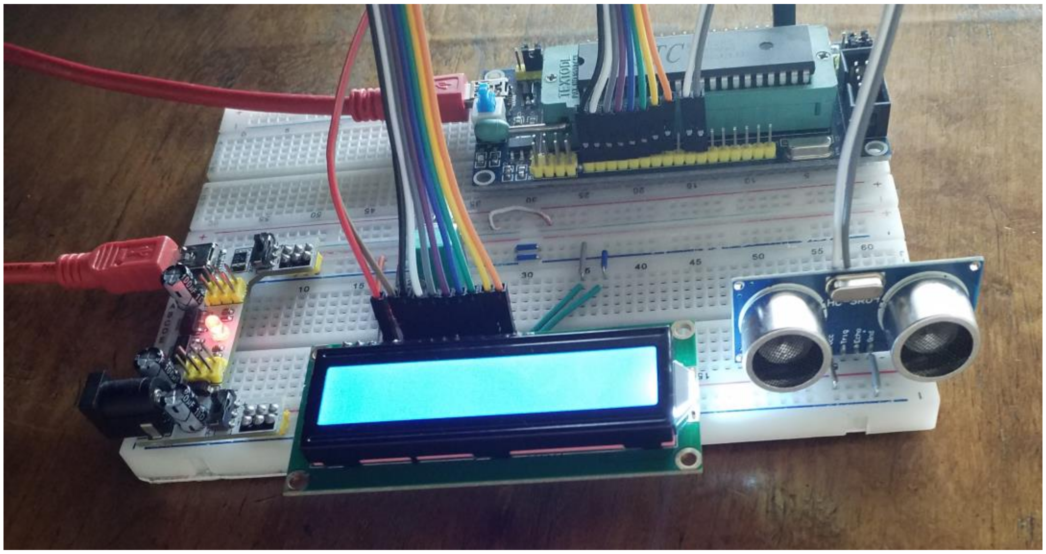


Figure 38: Physical circuit:

When powered on the LCD first displays some preliminary messages before taking the first measurement:

1. Message 1

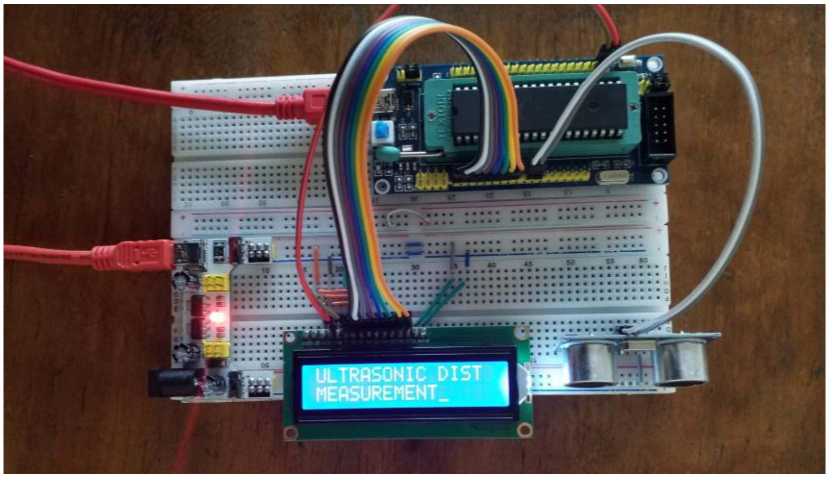


Figure 39: Circuit displays name of project

2. Message 2

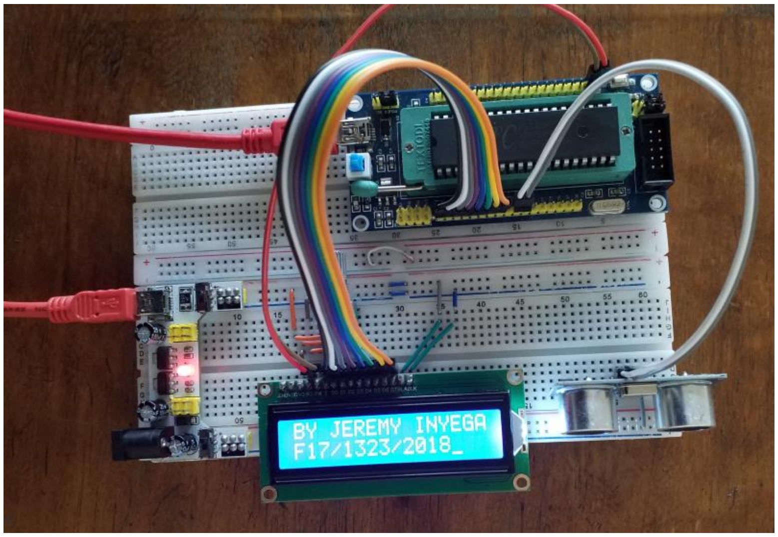


Figure 40: Circuit displays project author

3. Measurement

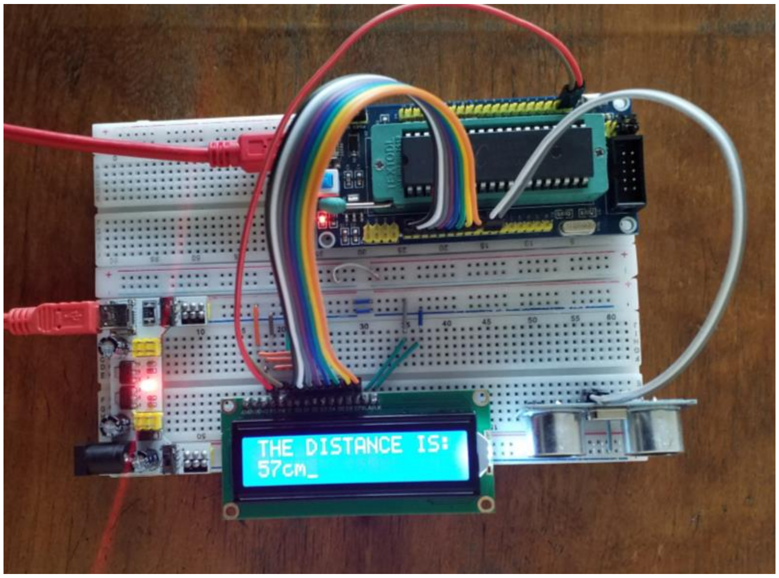


Figure 41: Circuit displays measured distance

CHAPTER 5: RESULTS AND ANALYSIS

5.1. SIMULATIONS

When running simulation the following trend emerged:

- The system experiences no errors between 0 and 27 cm
- The system experiences 1 cm error between 28 and 142cm
- The system experiences 2 cm error between 143 and 377 cm
- The system experiences 3 cm error between 378 cm and above

Table 2: Simulation Mean Square Error

Actual Distance	Simulation	Error Squared
2	2	0
10	10	0
40	41	1
80	81	1
120	121	1
160	162	4
200	202	4
240	242	4
280	282	4
320	322	4
360	362	4
400	403	9
Mean Error		3

5.2. REAL WORLD TESTING

Table 3: Physical system mean squared error

Actual Distance	**Indoor test**	**Test 1**	**Test 2**	**Test 4**	**Test average**	**Error Squared**
2	2	2	2	2	2	0
10	10	11	11	11	10.75	0.5625
40	41	40	41	41	40.75	0.5625
80	80	80	80	80	80	0
120	121	121	120	121	120.75	0.5625
160	160	157	161	161	159.75	0.0625
200	201	202	202	202	201.75	3.0625
240	240	241	242	241	241	1
280	281	280	280	280	280.25	0.0625
320		322	322	321	321.6666667	2.777777778
360		361	361	360	360.6666667	0.444444444
400		403	402	402	402.3333333	5.444444444
440		441	442	442	441.6666667	2.777777778
Mean Error						1.332264957

5.3. RESULTS ANALYSIS

It was noted that the maximum usable range was around 455 cm. Though, the system displayed values up to 503 cm, such values were difficult to replicate in standard conditions showing that possibly the HC-SR04 device needs some fine tuning to detect longer measures.

The system performed in line with the simulated expectations and event obtained measurements closer to the actual, though it may be due to positioning of the system. This is because the system rounds up to the nearest centimeter thus if the system displays 10 cm, it is possible that the object is actually at 9.1 cm. Alternatively, it may be due to parameters hardcoded to the simulation software, which do not take into account real world conditions such as variations in speed of sound.

Nevertheless, the system can be deemed to perform within the experimental tolerances (~1cm error).

CHAPTER 6: CONCLUSION AND RECOMMENDATIONS

6.1. CONCLUSION

From the objectives in chapter 1, the system was designed, implemented and its accuracy tested. The system is accurate within expectations in the design phase.

It was interesting to note that the actual system had better performance than the simulations but this may be due to difference in real vs virtual applications.

Performance during various weathers conditions was not tested, as it would prove hazardous or time consuming to find a day and / or place with the desired conditions.

6.2. FURTHER WORK

With the help of more powerful processors, especially those that can handle floating point math, it is possible to refine the design of this ultrasonic distance measurement system to obtain measurements that are more accurate. Further work on this project, swapping out the microcontroller may yield promising results.

6.3. RECOMMENDATION

This project demonstrates that ultrasonic distance measurement is accurate to within a few centimeters hence is suitable for applications where the general location of an object is required such as parking and not for precise measurements with low tolerances. It may be that more advanced devices exist that can fare better in the millimeter measurement range but for the implemented system, it rounds millimeters up to the nearest centimeter meaning a ~1 cm error is always expected.

REFERENCES

[1] D. Ensminger and L. J. Bond, Ultrasonics Fundamentals, Technologies and Applications, Boca Raton: CRC Press, 2012.

[2] E. J. Morgan, "HCSR04 Ultrasonic Sensor," 2014.

[3] STC, "STC89C51RC Features," [Online]. Available: https://www.stcmicro.com/datasheet/STC89C51RC_Features.pdf. [Accessed 7 6 2023].

[4] A. Sutar, P. Nekar, A. Terdale and M. M. Kolap, "Ultrasonic Radar Using 8051 Microcontroller," *International Journal of Research in Engineering and Science (IJRES),* vol. 9, no. 8, pp. 1-4, 2021.

[5] M. A. Mazidi and J. G. Mazidi, The 8051 Microcontroller and Embedded Systems, Chicago: Prentice Hall, 2000.

[6] Y. Girish, A. K. Jangir, A. K. Sharma and P. Kumar, "Microprocessor Based Ultrasonic Radar," *",International Journal of Engineering and Management Research*, vol. 7, no. 2, 2017.

[7] A. K. Shrivastava, A. Verma and S. P. Singh, "Distance Measurement of and Object or Obstacle by Ultrasonic Sensors," *International Journal of Computer Theory and Engineering*, vol. 2, no. 1, 2010.

[8] C. P. Kumar, R. K. Ghumman, C. D'souza and U. S. Pushpa, "Ultrasonic Range Finder using 8051," *International Journal for Research in Applied Science & Engineering Technology (IJRASET),* vol. 6, no. 1, 2018.

[9] P. Horowitz and W. Hill, "MICROCONTROLLERS," in *THE ART OF ELECTRONICS*, New York, Cambridge University Press, 2015, pp. 1053-1096.

[10] P. Scherz and S. Monk, "Microcontrollers," in *Practical Electronics For Inventors*, New York, McGraw-Hill Education , 2016, pp. 843-895.

[11] Pepperl+Fuchs, Ultrasonic Sensors, Germany: Pepperl+Fuchs, 2005.

[12] Ceptimus, "Breadboard STC89C52 programming," 3 1 2016. [Online]. Available: https://youtu.be/6cBVH6XN3Bw. [Accessed 6 6 2023].

[13] Ceptimus, "BreadboardSTC89.pdf," 3 1 2016. [Online]. Available: https://ceptimus.co.uk/BreadboardSTC89.pdf. [Accessed 6 6 2023].

[14] Ceptimus, "ISP programmer for the STC89C51 and STC89C52," 15 12 2015. [Online]. Available: https://ceptimus.co.uk/index.php/2015/12/15/isp-programmer-for-the-stc89c51-and-stc89c52/. [Accessed 6 6 2023].

[15] ELECTROGINEER, "Ultrasonic Range Finder Using 8051," [Online]. Available: https://www.instructables.com/Ultrasonic-Range-Finder-Using-8051/. [Accessed 6 6 2023].

[16] "Ultrasonic Module HC-SR04 Interfacing with 8051," [Online]. Available: https://www.electronicwings.com/8051/ultrasonic-module-hc-sr04-interfacing-with-8051. [Accessed 6 6 2023].

[17] "LCD16x2 Interfacing in 8-bit with 8051," [Online]. Available: https://www.electronicwings.com/8051/lcd16x2-interfacing-in-8-bit-with-8051. [Accessed 6 6 2023].

[18] Marlin P. Jones and Associates, "HC-SR04 Ultrasonic Ranging for Arduino (Datasheet)," [Online]. Available: https://www.mpja.com/HC-SR04-Ultrasonic-Ranging-for-Arduino/productinfo/19605+UT/. [Accessed 6 6 2023].

[19] M. A. Breazeale, J. S. Heyman, J. H. Cantrell, Jr and P. D. Edmonds, "ULTRASONIC WAVE VELOCITY AND," in *Methods of Experimental Physics : Volume 19 - Ultrasonics*, Menlo Park, Academic Press, 1981, pp. 67-137.

[20] Intel Corp, MCS-51 8- Bit Control Oriented Microcomputers, 1988.

[21] Phillips Semiconductors, Datasheet: 80C51 8-bit microcontroller family, 2000.

[22] STC, "STC89C51RC Datasheet," [Online]. Available: https://www.stcmicro.com/datasheet/STC89C51RC-en.pdf. [Accessed 7 6 2023].

[23] "LCD 16×2 Pinout, Commands, and Displaying Custom Characters," 5 1 2022. [Online]. Available: https://www.electronicsforu.com/technology-trends/learn-electronics/16x2-lcd-pinout-diagram. [Accessed 6 6 2023].

[24] M. B. FOLLOWR, "Basic Character LCD Hookup Guide," 16 5 2019. [Online]. Available: https://learn.sparkfun.com/tutorials/basic-character-lcd-hookup-guide/all. [Accessed 6 6 2023].

[25] Xianmen Amotec Display Co. Ltd,, "Specifications of LCD Module," 2008.

[26] J. Blitz, Ultrasonics: Methods and Applications, London: Butterworth & Co, 1971.

[27] STC, "STC89C51RC series," STC, 2014-2022. [Online]. Available: https://www.stcmicro.com/stc/stc89c51rc.html. [Accessed 6 6 2023].

[28] K. Bhatia and A. Pathak, "Factors Affecting Accuracy of Distance Measurement System Based on Ultrasonic Sensor in Air," *International Journal of Recent Technology and Engineering (IJRTE),* vol. 8, no. 2S11, pp. 2143-2144, 2019.

[29] Texas Instruments Incorporated, "Ultrasonic Distance Measurement using the TLV320AIC3268 miniDSP CODEC TIDA-00403," Texas Instruments Incorporated, Dallas, 2015.

[30] Senix distance and level sensors, "Ultrasonic Sensor Accuracy," [Online]. Available: https://senix.com/ultrasonic-sensor-accuracy/. [Accessed 6 6 2023].

[31] Sensor Partners, "Detection range of ultrasonic sensors," Sensor Partners, [Online]. Available: https://sensorpartners.com/en/kennisbank/detection-range-of-an-ultrasonic-sensor/. [Accessed 6 6 2023].

[32] MaxBotix, "Ultrasonic Sensors: Advantages and Limitations," MaxBotix, 11 9 2019. [Online]. Available: https://maxbotix.com/blogs/blog/advantages-limitations-ultrasonic-sensors. [Accessed 6 6 2023].

APPENDIX A: CODE

```
ORG 0000H
LJMP START


 ORG 001BH
 INC A; increments A for every timer1 overflow
 JNZ EXIT
 MOV R2, #0FFH ; Accumulator overflow check value
 EXIT: RETI



 ORG 0030H
 START:MOV P1,#00000000B   ; sets P1 as output port
 CLR P3.0                  ; sets P3.0 as output for sending trigger
 SETB P3.1                 ; sets P3.1 as input for receiving echo
 MOV TMOD,#00100000B       ; sets timer1 as mode 2 auto reload timer
 MOV IE, #10001000B        ; timer 1 overflow interrupt active

 ACALL INIT

 ACALL Disp_MSG1
 ACALL DELAY2
 ACALL Disp_MSG2
 ACALL DELAY4

 ACALL CLSCR

 ACALL Disp_MSG3
 ACALL DELAY2
 ACALL Disp_MSG4
 ACALL DELAY4

 ACALL CLSCR




 MAIN:
 MOV TL1,#-53      ; loads the initial value to start counting from (54 count = 1cm)
                   ; reduced to 50 to account for processing cycles
 MOV TH1,#-53      ; loads the reload value
                   ; use maximum 54 for 11.0592 MHZ and 58 for 12MHZ
 MOV A,#00000000B ; clears accumulator

 SETB P3.0         ; starts the trigger pulse
 ACALL DELAY1      ; gives 10uS width for the trigger pulse
 CLR P3.0          ; ends the trigger pulse

 HERE:JNB P3.1,HERE ; loops here until echo is received
 BACK:SETB TR1       ; starts the timer1
 HERE1:JNB TF1,HERE1 ; loops here until timer overflows (54 count = 1cm)
 JB P3.1,BACK      ; jumps to BACK if echo is still available.
 CLR TR1           ; stops timer
 MOV R0,A          ; saves the value of A to R0


 ACALL CLSCR
 ACALL Disp_MSG5
 ACALL DELAY2
 ACALL DISLOOP     ; calls the display loop
 ACALL DELAY3

 SJMP MAIN         ; jumps to MAIN loop




 DISLOOP:
```

```
MOV A, #0C0H        ; force cursor to second line
ACALL initial

ACALL DELAY2
MOV DPTR, #VALUE    ;value lut location

MOV A,R0            ; set A to value in R0


CJNE R2, #0FFH, NORMAL; for values over 255
;Isolating Hundreds
MOV B, #100D
DIV AB
MOV R2, A
;Isolating tens
MOV A, B
MOV B, #10D
DIV AB
MOV R3, A
MOV R4, B

MOV A, R4
ADD A, #5D
MOV B, #10D
DIV AB
MOV R4, B

ADD A, R3
ADD A, #5D
MOV B, #10D
DIV AB
MOV R3, B

ADD A, R2
ADD A, #2D
MOV R2, A


;Display Values
MOV A, R2
ACALL Digit_display
MOV A, R3
ACALL Digit_display
MOV A, R4
ACALL Digit_display

MOV R2, #0000H

SJMP UD ; jump to display cm

NORMAL: ; values below 255
MOV B, #100D
DIV AB
JZ TENS
ACALL Digit_display
;Isolating tens
TENS: MOV A, B
MOV B, #10D
DIV AB
ACALL Digit_display
;Isolating ones
MOV A, B
ACALL Digit_display

UD: MOV DPTR ,#UNITS    ; display cm routine
MOV A,#0D
ACALL Digit_display
MOV A,#1D
ACALL Digit_display
```

```
RET

Initial: ACALL DELAY2 ; initialization subroutines
MOV P1, A
CLR P2.0
SETB P2.1
ACALL DELAY2
CLR P2.1
RET

Digit_display: ACALL DELAY2
MOVC A,@A+DPTR    ; gets the digit drive pattern for the content in A
MOV P1, A
SETB P2.0
SETB P2.1
CLR P2.1
RET

DELAY1: MOV R6,#4D     ;~10uS delay
L: DJNZ R6,L
RET


DELAY2: MOV R6, #4D  ;~1 ms delay
L1: MOV R7, #125D
L2: DJNZ R7, L2
DJNZ R6, L1
RET


DELAY3:MOV R5, #125D   ; ~10 sec delay
D1: MOV R6, #200D
D2: MOV R7, #200D
INNER: DJNZ R7,INNER
DJNZ R6, D2
DJNZ R5, D1
RET



DELAY4:MOV R5, #150D   ; ~ 3 sec delay
D14: MOV R6, #100D
D24: MOV R7, #100D
INNER4: DJNZ R7,INNER4
DJNZ R6, D24
DJNZ R5, D14
RET



Disp_MSG1:
MOV A, #080H
ACALL Initial


MOV DPTR, #MESSAGE1
MOV R1, #15D

MSG1: MOV A,#00H
ACALL Digit_display
INC DPTR
DJNZ R1, MSG1

RET


Disp_MSG2:
MOV A, #0C0H
```

```
ACALL Initial


MOV DPTR, #MESSAGE2
MOV R1, #11D

MSG2: MOV A,#00H
ACALL Digit display
INC DPTR
DJNZ R1, MSG2

RET

Disp_MSG3:
MOV A, #80H
ACALL Initial


MOV DPTR, #MESSAGE3
MOV R1, #16D

MSG3: MOV A,#00H
ACALL Digit_display
INC DPTR
DJNZ R1, MSG3

RET

Disp_MSG4:
MOV A, #0C0H
ACALL Initial


MOV DPTR, #MESSAGE4
MOV R1, #13D

MSG4: MOV A,#00H
ACALL Digit display
INC DPTR
DJNZ R1, MSG4

RET


Disp_MSG5:
MOV A, #080H
ACALL Initial


MOV DPTR, #MESSAGE5
MOV R1, #16D

MSG5: MOV A,#00H
ACALL Digit_display
INC DPTR
DJNZ R1, MSG5

RET




INIT:MOV A, #38H   ; use 2 lines and 5*7
ACALL initial
ACALL DELAY2
MOV A, #0EH   ;cursor blinking off
ACALL initial
ACALL DELAY2
ACALL CLSCR
```

```
RET


CLSCR:MOV A, #01H   ;clear screen
ACALL initial
ACALL DELAY2
RET




VALUE:DB '0'
      DB '1'
      DB '2'
      DB '3'
      DB '4'
      DB '5'
      DB '6'
      DB '7'
      DB '8'
      DB '9'

UNITS: DB 'c'
       DB 'm'

MESSAGE1: DB 'ULTRASONIC DIST',0
MESSAGE2: DB 'MEASUREMENT',0
MESSAGE3: DB 'BY JEREMY INYEGA',0
MESSAGE4: DB 'F17/1323/2018'
MESSAGE5: DB 'THE DISTANCE IS:',0

END
```

APPENDIX B: BILL OF MATERIALS

Table 4: Bill of Materials

Item No.	Item	Cost per unit (KSh)	Units	Item cost (KSh)	Place purchased
1	STC89C52Rc chip	200	1	200	Pixel electric
2	HC-SR04 Ultrasonic Module	200	1	200	Ask Electronics
3	16 x 2 LCD	350	1	350	Ask Electronics
4	USB to Serial adapter	400	1	400	Ask Electronics
5	Bread board	200	1	200	Ask Electronics
6	Breadboard power supply	150	1	150	Ask Electronics
7	Power supply cables	100	2	200	Ask Electronics
8	STC89C52RC seat	517	1	517	Jumia
9	Male to female wires	115	1	115	Ask Electronics
10	Potentiometer	50	1	50	Ask Electronics
11	Header pins	20	1	20	Ask Electronics
12	Connecting wires	350	1	350	Ask Electronics
TOTAL COST				2752	

Note:
Some materials are sold as a large bunch hence item cost may actually be lower.
These prices are accurate to time of purchase and subject to change based on seller preferences.
Prices exclude shipping.

APPENDIX C: POSSIBLE LCD CHARACTERS