SMART Home Project

_

PROJECT DOCUMENTATION

PREPARED BY:

Jomana Ashraf

PREPARED FOR:

AMIT

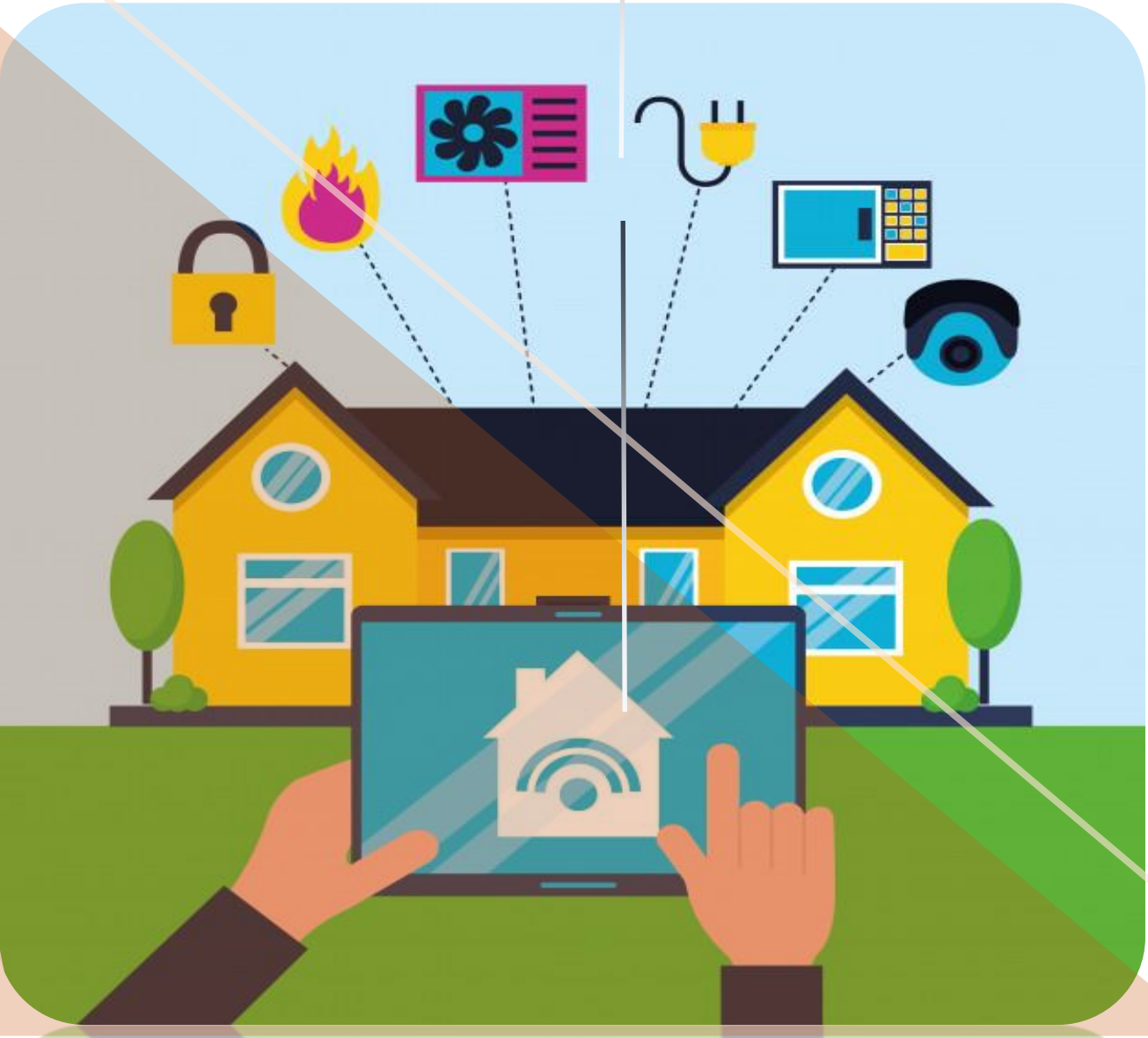


Table of Contents

List of Figures

1 INTRODUCTION

1.1 What are Smart Homes?

A smart home refers to a convenient home setup where appliances and devices can be automatically controlled remotely from anywhere with an internet connection using a mobile or other networked device. Devices in a smart home are interconnected through the internet, allowing the user to control functions such as security access to the home, temperature, lighting, and a home theater remotely.

1.2 How do Smart Homes work?

A smart home's devices are connected with each other and can be accessed through one central point—a smartphone, tablet, laptop, or game console. Door locks, televisions, thermostats, home monitors, cameras, lights, and even appliances such as the refrigerator can be controlled through one home automation system. The system is installed on a mobile or other networked device, and the user can create time schedules for certain changes to take effect.

Smart home appliances come with self-learning skills so they can learn the homeowner's schedules and make adjustments as needed. Smart homes enabled with lighting control allow homeowners to reduce electricity use and benefit from energy-related cost savings. Some home automation systems alert the homeowner if any motion is detected in the home when they're away, while others can call the authorities—police or the fire department—in case of imminent situations. Once connected, services such as a smart doorbell, smart security system, and smart appliances are all part of the internet of things (IoT) technology, a network of physical objects that can gather and share electronic information.

2 CIRCUIT

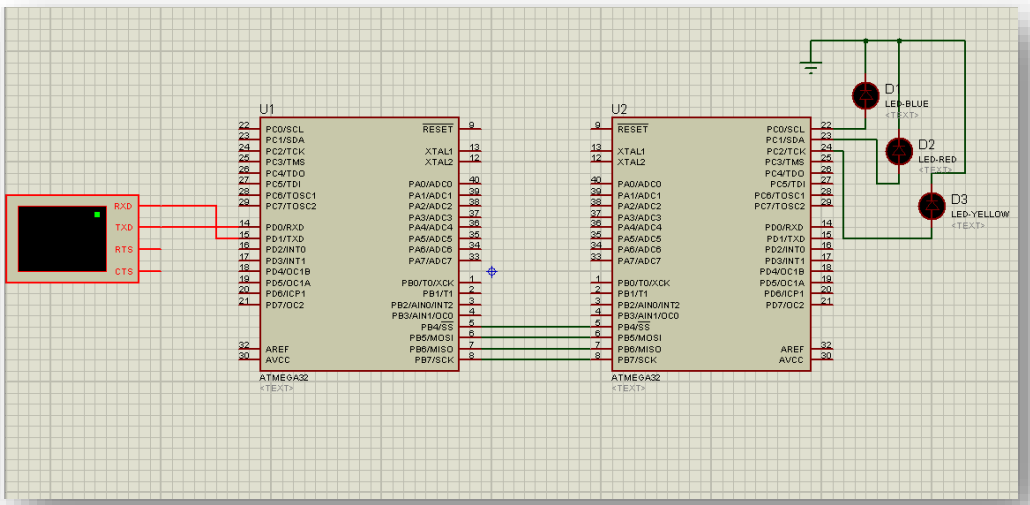
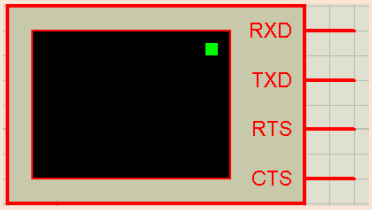
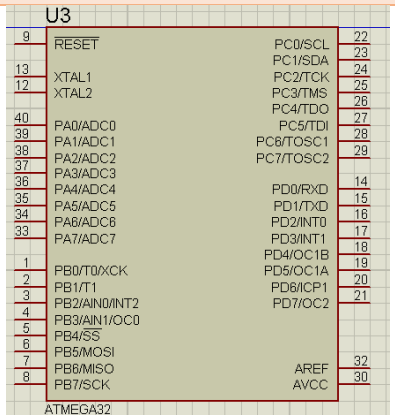


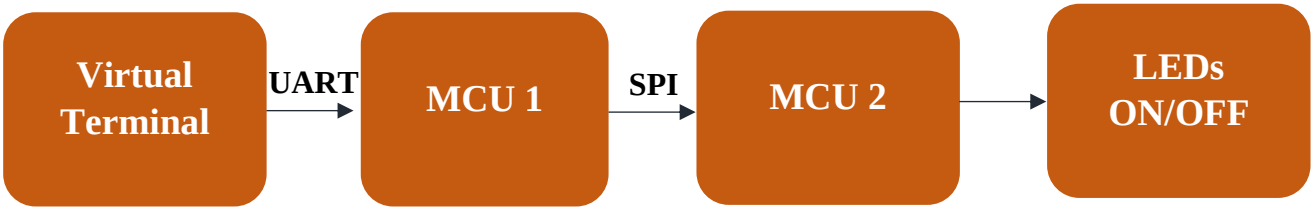
Figure 1, Smart Home Circuit

2.1 Components of the Circuit

Component	Quantity	Description
	**1**	Virtual Terminal used as the Bluetooth Module.
	**2**	At mega 32 is the microcontroller of the system
	3	Led

2.2 Circuit Explanation

The virtual terminal is used as a Bluetooth module; it is connected to the master microcontroller so; the Tx is attached to the Rx of the MCU and the Rx is connected to the Tx of the MCU. The communication protocol between the first MCU and the virtual terminal is UART. Although, there is another MCU which is the slave one that is connected to the first MCU and 3 LEDs. The communication protocol between the two MCUs is SPI Protocol. If **"A"** is written in the virtual terminal; the master sends it to the slave to make it operate the blue led. If **"B"** is used; so, the red led is ON. If **"C"** is written, the yellow led is executed. To make all the LEDs off; you can write **"S"** in the virtual terminal.



3 CODE

3.1 Master Flowchart

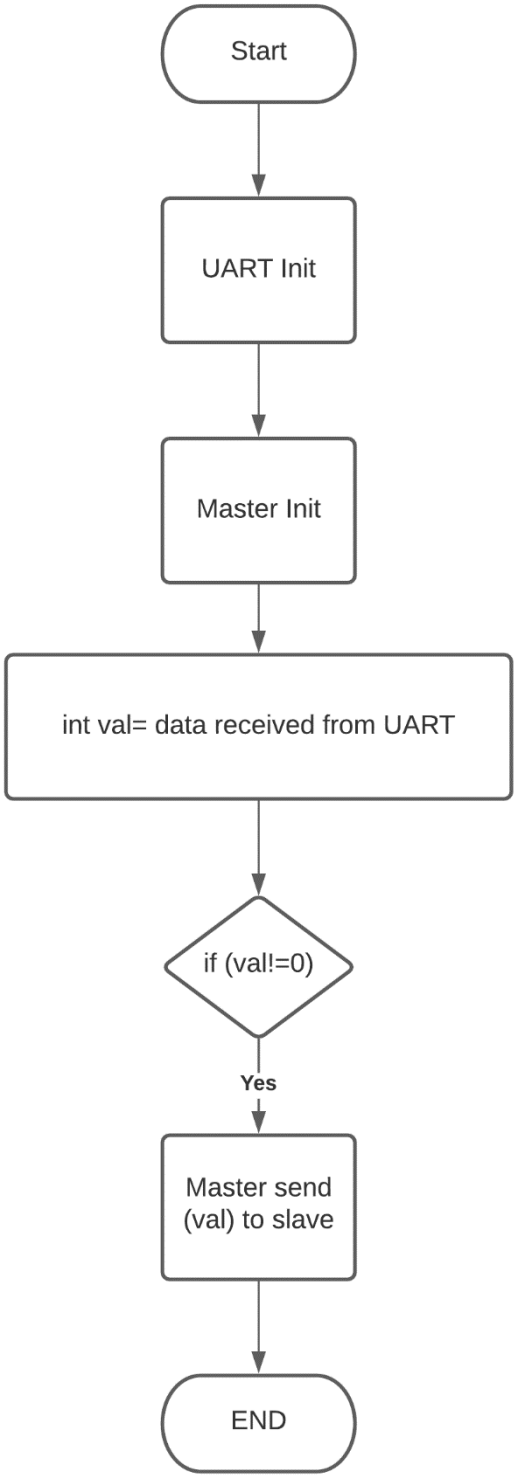


Figure 2, Mater Flowchart

3.2 Slave Flowchart

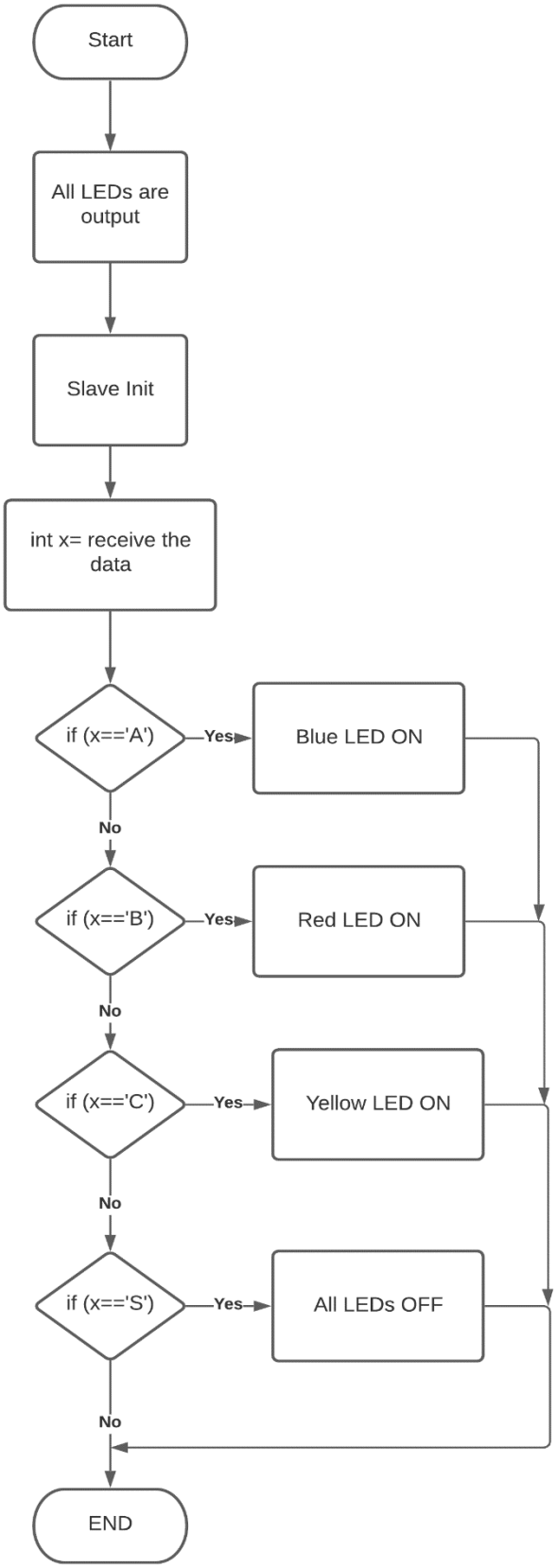


Figure 3, Slave Flowchart

4 APPENDIX

4.1 Master Code

```
#include "STD_TYPES.h"
#include "BIT_MATH.h"

#include "MUSART_Interface.h"
#include "MSPI_Interface.h"
#include "MDIO_Interface.h"

int main(void)
{
	MUSART_VidInit(9600);
	MCAL_SPIMasterInit();
	u8 val;

	while(1)
	{
		val = MUSART_VidReceiveDataPolling();
		if (val!=0)
		{
			MCAL_SPIMasterSend(val);
		}
	}
	return 0;
}
```

4.2 Slave Code

```
#include "STD_TYPES.h"
#include "BIT_MATH.h"

#include "MSPI_Interface.h"
#include "MDIO_Interface.h"
#include "LCD_Interface.h"

int main(void)
{
	MDIO_VidSetPinDir(PortC,Pin0,MDIO_OUTPUT);
	MDIO_VidSetPinDir(PortC,Pin1,MDIO_OUTPUT);
	MDIO_VidSetPinDir(PortC,Pin2,MDIO_OUTPUT);
	MCAL_SPISlaveInit();
	u8 x;
	while (1)
	{
		x= MCAL_SPISlaveReceive(0);
		if(x=='A')
		{
			MDIO_VidSetPinValue(PortC,Pin0,MDIO_HIGH);
		}
		else if (x=='B')
		{
			MDIO_VidSetPinValue(PortC,Pin1,MDIO_HIGH);
		}
		else if (x=='C')
		{
			MDIO_VidSetPinValue(PortC,Pin2,MDIO_HIGH);
```

```
                }
                else if (x=='S')
                {
                        MDIO_VidSetPinValue(PortC,Pin0,MDIO_LOW);
                        MDIO_VidSetPinValue(PortC,Pin1,MDIO_LOW);
                        MDIO_VidSetPinValue(PortC,Pin2,MDIO_LOW);
                }
        }
        return 0;
}
```