UNIVERSITY OF TEXAS ARLINGTON

EE-5314

EMBEDDED MICROCONTROLLER SYSTEMS

FINAL PROJECT REPORT

TWO-WAY COMMUNICATION BETWEEN SYSTEMS

(INFRARED TX AND RX CIRCUITS)

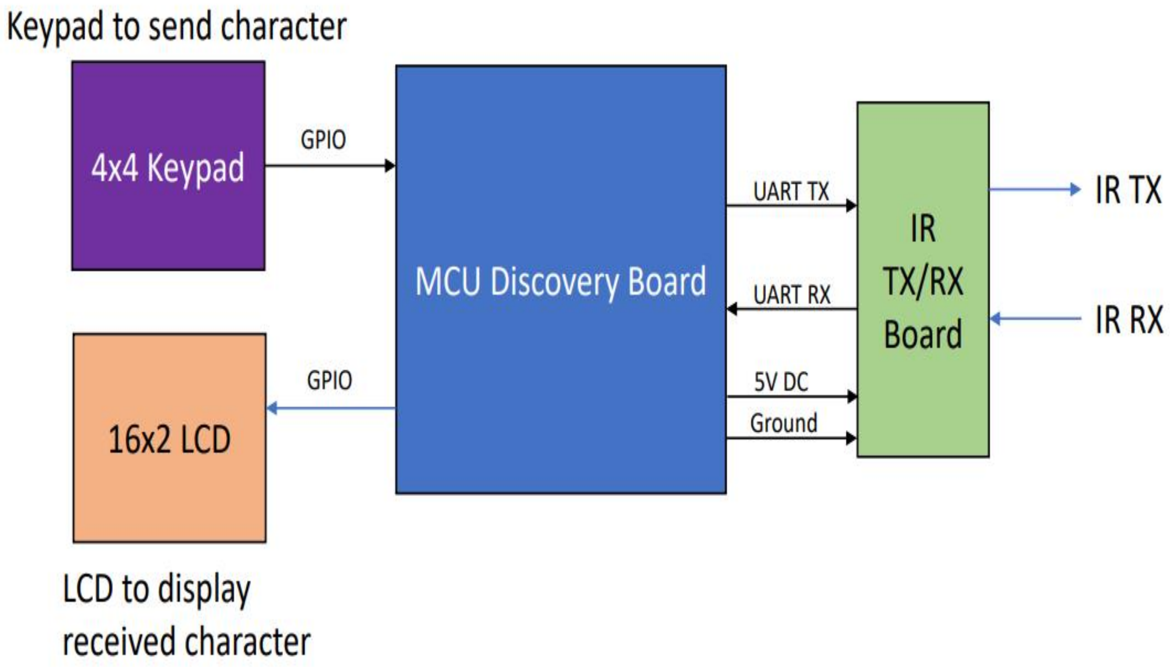
Done by

PRAJWAL GOWDA RAMPURA SURESH 1002064165

Problem Statement

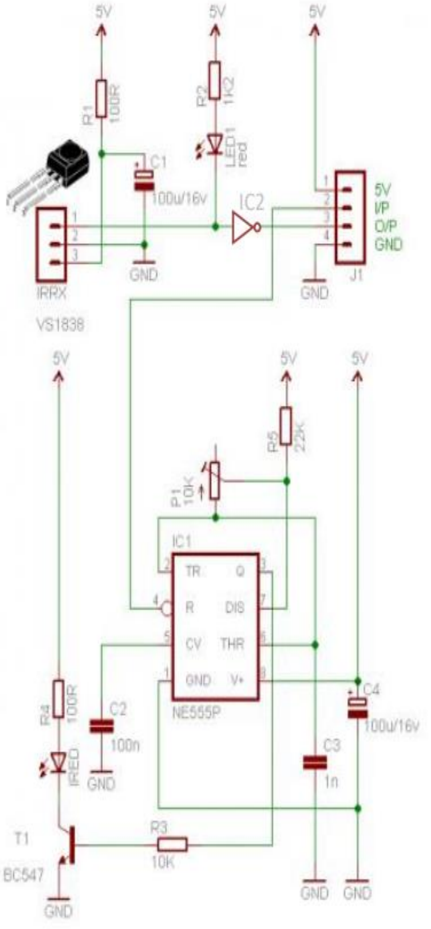
The goal of our project is to establish a two-way communication between two systems. The two systems will be infrared based transmitter (Tx) and receiver (Rx) circuits. We are using STM32F407VGT6 microcontroller to perform this task. External peripherals include an external LCD display and external keypad.

Project High-Level Design



Infrared Tx/Rx Schematic Circuit

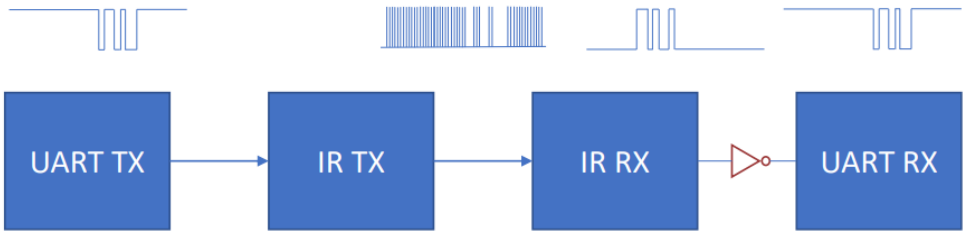
We are using USART interface to drive the IR TX and interpret the IR receive signal. The baud rate is set to 1200 bps, since we are using 38 kHz pulses and need enough pulses per transmit bit. There will be polarity reversal when the circuit is receiving pulses and we need an inverter to reverse this polarity change with in IR receiver. In this project we have placed the inverter after IR receiver as shown in the below circuit diagram. Here led 1 is lit when the IR receiver is receiving pulses and it is lit by default and it will confirm that our circuit has the active IR connection.



R1 — 100R ¼-W carbon resistor
R2 — 1K2 ¼-W carbon resistor
R3 — 10K ¼-W carbon resistor
R4 — 100R ¼-W carbon resistor
R5 — 22K ¼-W carbon resistor
P1 — 10K preset pot
C1&C4 — 100-uF/16-V electrolytic capacitor
C2 — 100-nF ceramic capacitor
C3 — 1-nF ceramic capacitor/tantalum capacitor
IRED — 5-mm infrared LED
IRRX — VS1838 (or TSOP1838) infrared receiver
LED1 — 5-mm red LED
T1 — BC547 NPN transistor
IC1 — NE555P timer chip
J1 — Four-pin male header
IC2 — Inverter

Circuit with Inverter placed after IR Receiver

Inverter location: Placed after IR Receiver (reverses the polarity)



Microcontroller Code Description and Approach

Microcontroller code for this project includes the following parts.

- GPIO driver development.
- USART driver development.
- LCD interface.
- Keypad interface.
- USART communication.

GPIO driver development

This driver file consists of all GPIO peripherals pin details, configurations, functions definitions like read, write, toggle and so on. Some of the functions are listed below.

- `void GPIO_PeriClockControl(GPIO_RegDef_t *pGPIOx, uint8_t EnorDi);`
- `void GPIO_Init(GPIO_Handle_t *pGPIOHandle);`
- `void GPIO_DeInit(GPIO_RegDef_t *pGPIOx);`
- `uint8_t GPIO_ReadFromInputPin(GPIO_RegDef_t *pGPIOx, uint8_t PinNumber);`
- `uint16_t GPIO_ReadFromInputPort(GPIO_RegDef_t *pGPIOx);`
- `void GPIO_WriteToOutputPin(GPIO_RegDef_t *pGPIOx, uint8_t PinNumber, uint8_t Value);`
- `void GPIO_WriteToOutputPort(GPIO_RegDef_t *pGPIOx, uint16_t Value);`
- `void GPIO_ToggleOutputPin(GPIO_RegDef_t *pGPIOx, uint8_t PinNumber);`

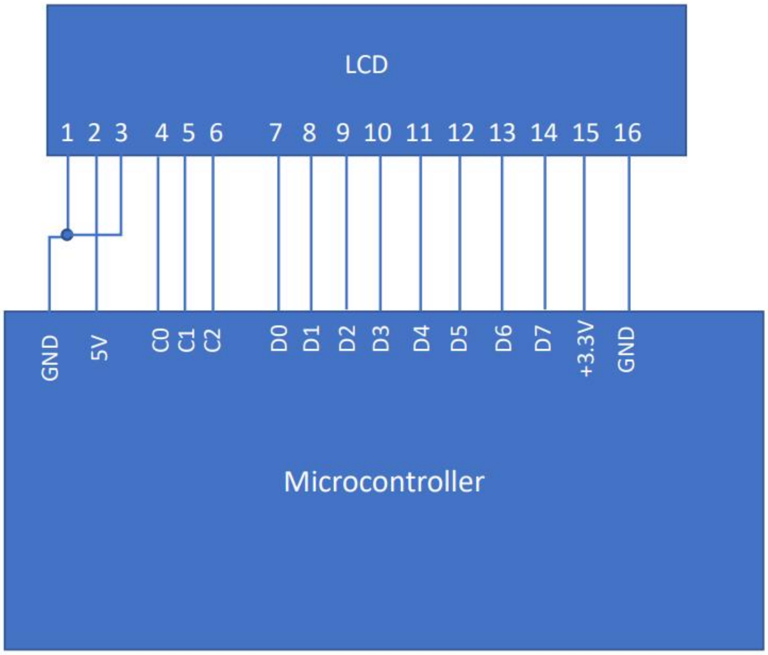
USART driver development

This driver file consists of all USART peripherals configuration details, functions definitions like send, receive etc. Some of the functions are listed below.

- `void USART_PeriClockControl(USART_RegDef_t *pUSARTx, uint8_t EnOrDi);`
- `void USART_Init(USART_Handle_t *pUSARTHandle);`
- `void USART_DeInit(USART_Handle_t *pUSARTHandle);`
- `void USART_SendData(USART_Handle_t *pUSARTHandle, uint8_t *pTxBuffer, uint32_t Len);`
- `void USART_ReceiveData(USART_Handle_t *pUSARTHandle,uint8_t *pRxBuffer, uint32_t Len);`
- `uint8_t USART_SendDataIT(USART_Handle_t *pUSARTHandle,uint8_t *pTxBuffer, uint32_t Len);`
- `uint8_t USART_ReceiveDataIT(USART_Handle_t *pUSARTHandle,uint8_t *pRxBuffer, uint32_t Len);`

LCD Interface

LCD Connection to MCU



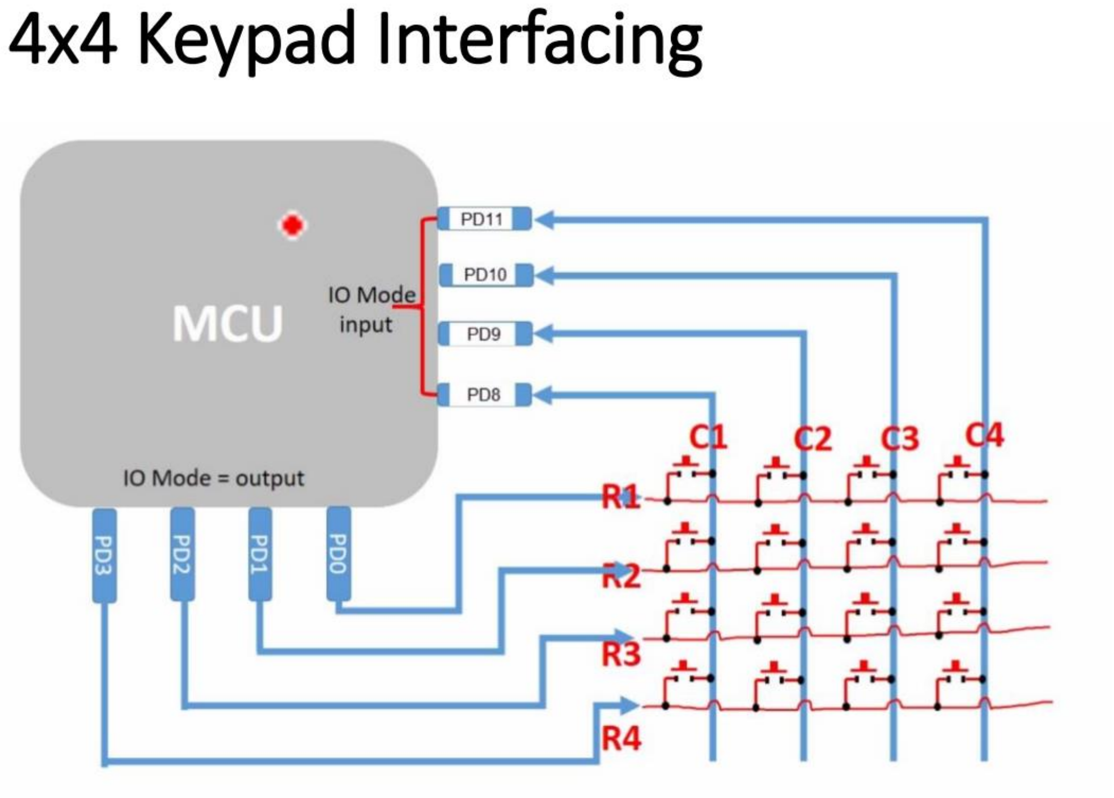
LCD pin definitions

PIN no.	Name
1	GND
2	VCC (+5V)
3	VEE
4	RS
5	R/W
6	E
7	D0
8	D1
9	D2
10	D3
11	D4
12	D5
13	D6
14	D7
15	LED+
16	LED-

Lcd interface command functions are present in this header file. Functions are:

- `void delayMs(uint32_t n);`
- `void LCD_command(unsigned char command);`
- `void LCD_Data(char data);`
- `void LCD_Init(void);`
- `void PORTS_Init(void);`
- `void Char_to_Binary(char data);`

Keypad Interface



Here, we make columns C1, C2, C3 and C4 pins connected to GPIO port D pins 9, 10, 11 and 12 as pull-up resistors and it will be input mode and R1, R2, R3 and R4 connected GPIO port D pins 0, 1, 2 and 3 as output mode. To detect exact button pressed in the keypad we will make that exact row and column low, and the pressed value will be detected.

Conclusion:

We have successfully demonstrated our working project establishing the communication between two systems IR transmitter and IR receiver. We’ve used inverter after IR receiver circuit for the demonstration. While doing this project, we have learned embedded microcontrollers concepts like GPIO drivers, USART drivers, external peripheral interface like keypad interface and lcd interface to the microcontroller.