The University of Texas at Arlington

CSE - 5400
Introduction to Computer Engineering
Wireless IR Communication

Submitted by: Rao Waqas Ali
Submitted to: Dr. Jason Losh and Nicholas Untrecht

Introduction

The goal of this project is to build a device capable of transmitting and receiving from multiple devices on a 40 kHz modulated IR link.

The software must support the following functionality:

1. UART Commands

The virtual COM port connected to UART0 will support these instructions:

Instructions	
send “string” send BYTE 0 [BYTE 1 […]]	Write one or more bytes to the TX FIFO feeding the the IR link
receive	Display the last received message in the RX FIFO
address ADD	Sets the address (ADD) of a node in EEPROM for subsequent use
set ADD, DATA	Sets the value of the data associated with a node’s address ADD
rgb ADD, R, G, B	Set the value of the {RGB} set associated with a node’s address ADD
get ADD	Gets the value of the data associated with a node’s address ADD
poll	Send a poll request to all devices on the bus
ack ON|OFF	Set the ACK setting to use in subsequent commands sent

Table 1

2. IR data format

A transmission requires a 9ms burst, followed by a 4.5ms pause, following by the following message bytes using the NEC modulation.

DST_ADD	The address to which the message is being sent
SRC_ADD	The address from which the message is being sent. This is used for the destination address for nodes acknowledging a message.
ACK|CMD	The ACK bit is set if a acknowledge is requested. The CMD field is 7 bits in length. Commands are Set (0x00), RGB (0x01), Get (0x10), Get response (0x11), Poll (0x20)
ARGS	Zero or more bytes of command-specific data: Set requires a DATA byte, RGB requires R, G, and B bytes, and Get response requires a DATA byte.
CHECKSUM	One's complement of the sum of the above bytes

Table 2

Theory of Operation

Uart0

There are three major sections of this code. The first one is the UART0 configuration and setting up of the UART0 to create a common terminal command interface. No C libraries was used to code the solution, allowing the entire interface to be coded in less than 1KiB of flash while using less than 128B of stackbased SRAM, no global variables, and no heap allocation. Not using the C library decreases the size of memory and also helps to mitigate multi-threaded C library reentrancy issues in RTOS solutions.

Transmission of IR signal

The PWM module with the help of timers was used to create a 50% duty cycle modulated signal, the interrupt service routine for both the timers and GPIN edge trigger were coded to send IR messages using an IR LED. The commands on terminal are used to send the data (data type explained in table 1)

Receiving the Data using IR Detector

An GPIN edge triggered ISR was coded to process the data received. Circular buffers were used to send and receive data. As the NEC protocol sends the data in binary format, whether to send LSB or MSB first was left on coder discretion. To make sure the correct was sent to the intended target, a packet was created to send data in sequence

described in table 2. Ack back from the target could be requested, however not always required.

Conclusion

The communication through NEC protocol is an useful and convenient way of handling devices. The simplicity of the protocol and less cost to make such a device is the reason that devices following this protocol are omnipresent in industry. However, coding the device as a semester project was challenging and tricky especially in case of lab 8, where I needed to identify the end of the variable length data.

The code is included in a separate file.

Thanks to the TA

I would like to take this opportunity to mention that the TA for this project has been really approachable and helpful. Without the help of the TA this project would not have been completed.