-need to limit current from voltage regulator (it usually outputs 1A - that is too much for us)

SOFTWARE NOTE:

-when we are programming the communication between the RF transmitter and receiver, to avoid noise, we should first send a signal like “DATA IS COMING” - when the receiver receives these bits, it will read a certain number of bytes of data (that we specify) and then stop reading data until it receives another “DATA IS COMING” bitstream.

LINKS:

RF Transmitter:

<https://www.sparkfun.com/products/10534>

RF Receiver:

<https://www.sparkfun.com/products/10532>

Quick Start guide for wireless transmission:

<https://www.sparkfun.com/datasheets/RF/KLP_Walkthrough.pdf>

Receiver/Transmitter GitHub Repository:

<https://github.com/sparkfun/RF_Links>

‘Keypad’ Scanning Helpful Links:

<https://books.google.com/books?id=5atwJG7D_HMC&pg=PA339&lpg=PA339&dq=12+key+keypad+9s12&source=bl&ots=KiZ_SDtxVH&sig=8eZBQbD2rqAkidB8Y5FOxbhgvjo&hl=en&sa=X&ved=0ahUKEwjM5bvm2aTJAhWFFR4KHWjDA8AQ6AEIHTAA#v=onepage&q=12%20key%20keypad%209s12&f=false>

Eagle Parts:

<http://www.elecrom.com/2009/10/09/eagle-library-list-of-most-commonly-used-electronics-components/>

Things to work on:

**TRANSMITTER SIDE:**

-the sound of the piezo buzzer will be adjusted using the PWM (output from PT3) (based on the potentiometer input on ATD channel 0 (AN0)

-RF Transmitter - how does it work? Uses the SCI - pin 1 - TX is the transmit data

-to power up the LED’s, we will use 2 outputs from the microcontroller: a clock (PT6) and data out (PT5), using the GAL 26V12 as the 12-bit shift register. This PLD has 12 outputs (we need to buy and program the GAL with Dataman). The clock should be transitioned every time we want to shift a bit out to update the status of the LED’s. This should be done for all 12 LED’s. NOTE: we are getting rid of the ‘in progress’ LED.

-the reset button is connected to ground on one side and pin 30 on the other side

-pushbuttons - We will use 7 pins (3 outputs (PT0-PT2), 4 inputs (AN1-AN4)). The 3 outputs (‘scan lines’) connect to the column lines and are controlled by the TIM module to switch between low and high. Whenever we change which column line is low, we then sample each of the 4 inputs from the row lines. If any of the row lines and column lines are both low, we can figure out which pushbutton was pressed. See the Lecture Notes picture in the drive for more info. The pull-up resistors are on the row outputs, which are normally high (hooked up to +5V). NOTE: we will have to do push-button debouncing in software (see Google Books link above)

Table for row/column lines mapped to the pushbuttons on the transmitter side:

|  |  |  |
| --- | --- | --- |
| **Button Pressed** | **Column Pin (3 total) (outputs)** | **Row Pin (4 total) (inputs)** |
| 1 Star | C1 (PT0) | R1 (AN1) |
| 2 Stars | C1 (PT0) | R2 (AN2) |
| 3 Stars | C1 (PT0) | R3 (AN3) |
| 4 Stars | C1 (PT0) | R4 (AN4) |
| 5 Stars | C2 (PT1) | R1 (AN1) |
| 1 Service | C2 (PT1) | R2 (AN2) |
| 2 Service | C2 (PT1) | R3 (AN3) |
| 3 Service | C2 (PT1) | R4 (AN4) |
| 1 Food | C3 (PT2) | R1 (AN1) |
| 2 Food | C3 (PT2) | R2 (AN2) |
| 3 Food | C3 (PT2) | R3 (AN3) |
| Submit | C3 (PT2) | R4 (AN4) |

-DONE: power jack (include a barrel jack input on the PCB)

**RECEIVER SIDE:**

-LCD (with backlight) w/ 8-bit shift register (connected as specified in Lab 8 - see lab document for how connections are done)

-reset button

-potentiometer/buzzer

-RF Receiver - 8 pins - pin 2 (RX - receive data) will be used

-microcontroller

-power jack

**-to be sent from transmitter to receiver:**

1. When the rating was input (upon reset, timer starts at 00:00:00 and counts up by seconds)
2. The Overall Rating
3. The Service Rating
4. The Food Rating

**PCB DESIGN NOTES:**

TRANSMITTER:

-we will have two 12-ribbon wire buses coming out:

-one will be 11 wires (8 buttons, GND, +5V, POT)

-the other will be 12 wires (1 for each LED)

-all of the grounds of the LED’s will be connected to eachother via the box (don’t need to pass through the breadboard - that would require too many wires)

-the piezo buzzer will be *on the board*

-TO\_LEDS pinout:

-pin 1: SERVICE LEFT

-pin 2: SERVICE MID

-pin 3: SERVICE RIGHT

-pin 4: FOOD LEFT

-pin 5: FOOD MID

-pin 6: FOOD RIGHT

-pin 7: STAR 1

-pin 8: STAR 2

-pin 9: STAR 3

-pin 10: STAR 4

-pin 11: STAR 5

-pin 12: COMPLETE

-TO\_BUTTONS pinout (see schematic for rows/columns explanation):

-pin 1: ROW 1 (TOP ROW)

-pin 2: ROW 2

-pin 3: ROW 3

-pin 4: ROW 4 (BOTTOM ROW)

-pin 5: COLUMN 3 (RIGHT COLUMN)

-pin 6: COLUMN 2

-pin 7: COLUMN 1 (LEFT COLUMN)

-pin 8: RESET

-pin 9: POT

-pin 10: unused

-pin 11: +5V

-pin 12: GND

RECEIVER:

-only needs 4 wires to come out: POT, +5V, GND, and RESET

-the piezo buzzer will be on the board

-OUTPINS pinout:

-pin 1: POT input

-pin 2: RESET button input

-pin 3: GND input

-pin 4: 5 V input

-pin 5: unused