EELE 465 Project 3 Demo

Erik and Zane

Project Overview

System Includes:

- 16-input Membrane Keypad used to:
  * Lock/unlock system
  * enter password to validate operations
  * Determine patterns and speed of phase changes on LCD display
- RGB LED informing user of system state
- LCD Display which displays one of a variety of patterns the user may choose
- Heartbeat LED to validate basic system operations

System Overview (Methods)

- main (orchestrates decision making and tracks current system state)
- read_input (reads an input from keypad)
- wait_for_unlock (while in locked state, controls changes of system state based upon user inputs)
- passkey (validates user's entered pin, when appropriate)
- update_color (updates current count-to variables to compare to timer ISR's pwms for RGB LED)
- patternX (an array of similar methods selected based upon user input which update LCD display phases of each available pattern)

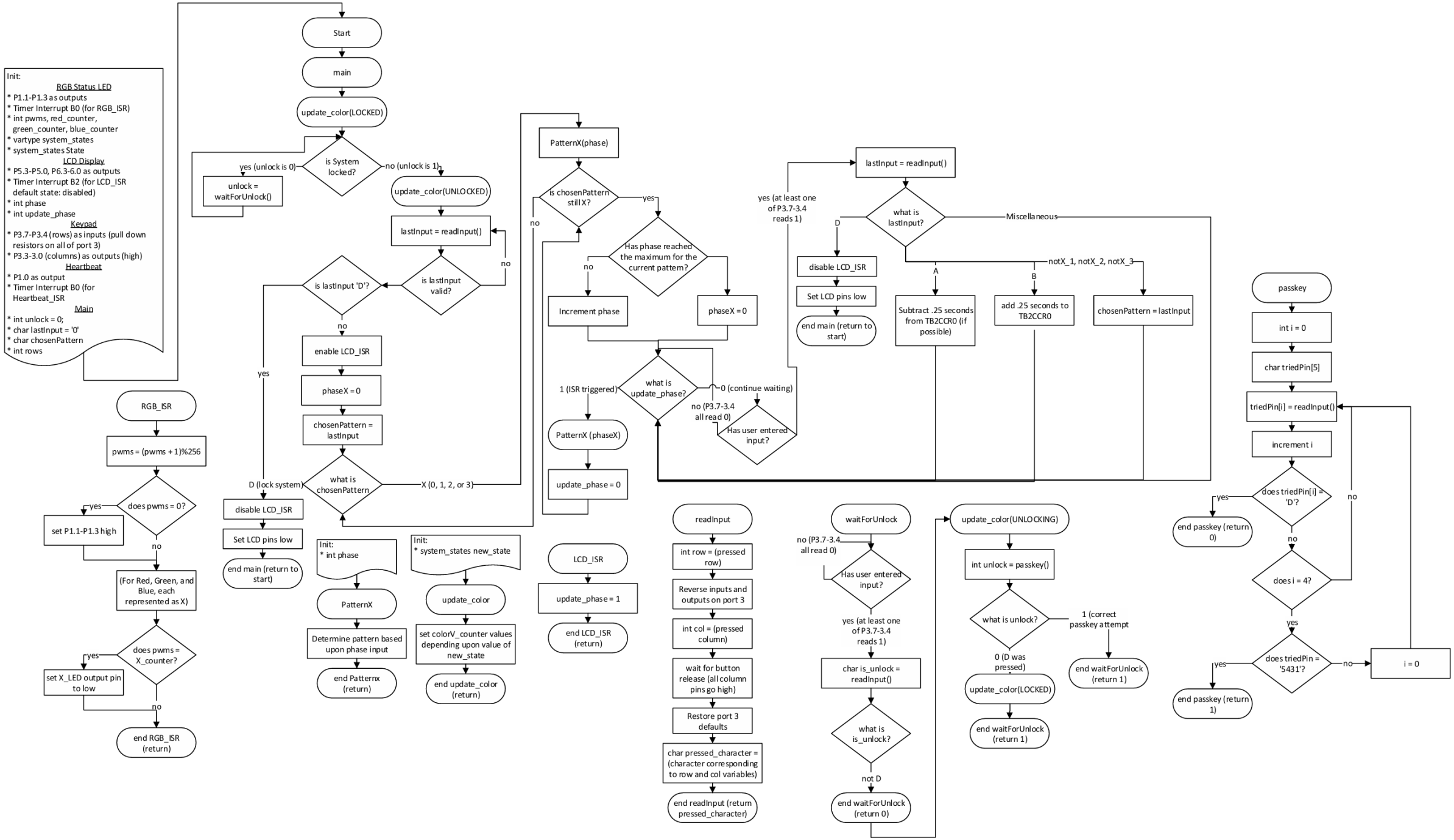
System Overview (Interrupts)

- RGB_ISR (controls red, green, and blue uptime at ~50 Hz)
- Heartbeat_ISR (controls heartbeat LED)
- LCD_ISR (while LCD is operating, controls when next pattern phase is to be sent)

* Extra*

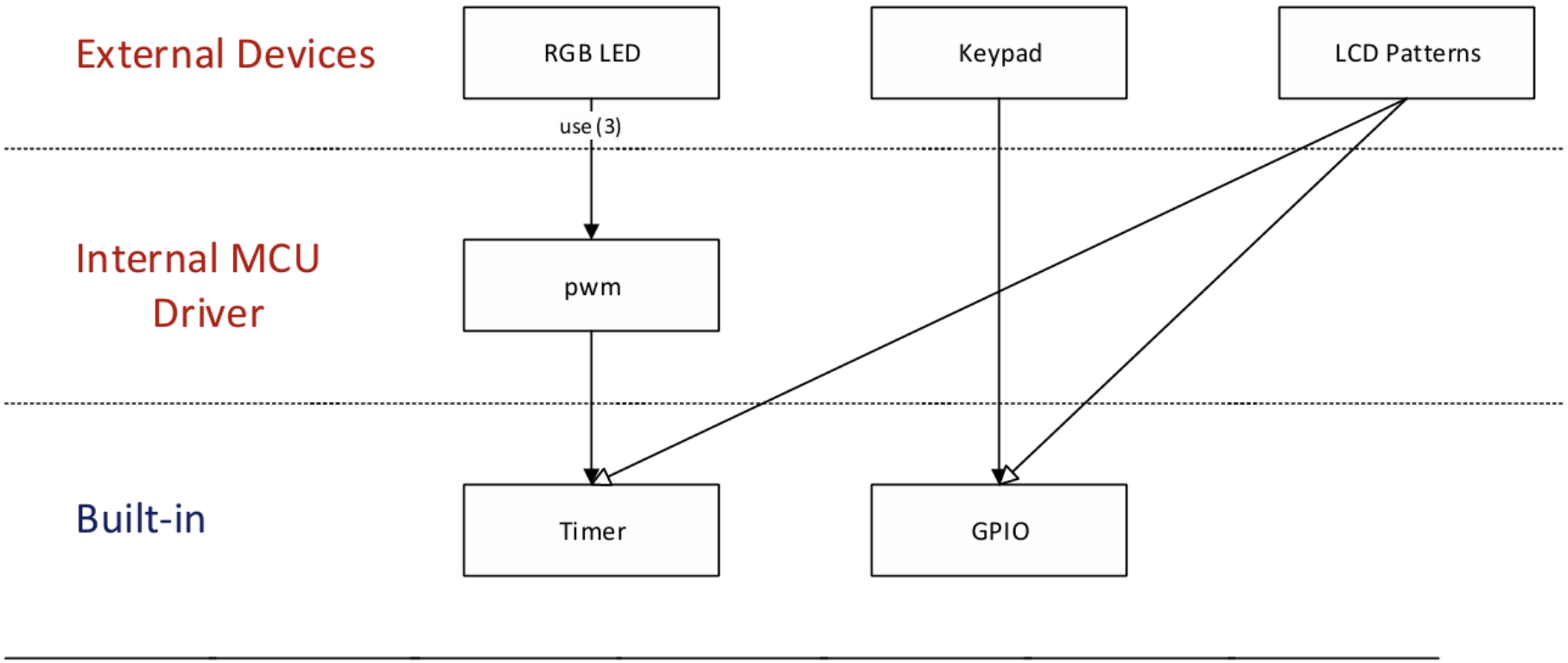
- Limit_ISR (determines whether time limit has been reached for passkey input)

(detailed flowchart next page)


Init:
RGB Status LED
* P1.1-P1.3 as outputs
* Timer Interrupt B0 (for RGB_ISR)
* int pwms, red_counter, green_counter, blue_counter
* vartype system_states
* system_states State
LCD Display
* P5.3-P5.0, P6.3-6.0 as outputs
* Timer Interrupt B2 (for LCD_ISR default state: disabled)
* int phase
* int update_phase
Keypad
* P3.7-P3.4 (rows) as inputs (pull down resistors on all of port 3)
* P3.3-3.0 (columns) as outputs (high)
Heartbeat
* P1.0 as output
* Timer Interrupt B0 (for Heartbeat_ISR
Main
* int unlock = 0;
* char lastInput = '0'
* char chosenPattern
* int rows
Start
main
update_color(LOCKED)
is System locked?
yes (unlock is 0)
unlock = waitForUnlock()
no (unlock is 1)
update_color(UNLOCKED)
lastInput = readInput()
is lastInput valid?
no
is lastInput 'D'?
yes
no
enable LCD_ISR
phaseX = 0
chosenPattern = lastInput
what is chosenPattern
D (lock system)
X (0, 1, 2, or 3)
disable LCD_ISR
Set LCD pins low
end main (return to start)
PatternX(phase)
is chosenPattern still X?
yes
no
Has phase reached the maximum for the current pattern?
no
yes
Increment phase
phaseX = 0
what is update_phase?
1 (ISR triggered)
0 (continue waiting)
no (P3.7-3.4 all read 0)
Has user entered input?
yes (at least one of P3.7-3.4 reads 1)
PatternX (phaseX)
update_phase = 0
lastInput = readInput()
what is lastInput?
D
Miscellaneous
disable LCD_ISR
Set LCD pins low
end main (return to start)
A
B
notX_1, notX_2, notX_3
Subtract .25 seconds from TB2CCR0 (if possible)
add .25 seconds to TB2CCR0
chosenPattern = lastInput
RGB_ISR
pwms = (pwms + 1)%256
does pwms = 0?
yes
set P1.1-P1.3 high
no
(For Red, Green, and Blue, each represented as X)
does pwms = X_counter?
yes
set X_LED output pin to low
no
end RGB_ISR (return)
Init:
* int phase
PatternX
Determine pattern based upon phase input
end Patternx (return)
Init:
* system_states new_state
update_color
set colorV_counter values depending upon value of new_state
end update_color (return)
LCD_ISR
update_phase = 1
end LCD_ISR (return)
readInput
int row = (pressed row)
Reverse inputs and outputs on port 3
int col = (pressed column)
wait for button release (all column pins go high)
Restore port 3 defaults
char pressed_character = (character corresponding to row and col variables)
end readInput (return pressed_character)
waitForUnlock
no (P3.7-3.4 all read 0)
Has user entered input?
yes (at least one of P3.7-3.4 reads 1)
char is_unlock = readInput()
what is is_unlock?
not D
end waitForUnlock (return 0)
update_color(UNLOCKING)
int unlock = passkey()
what is unlock?
0 (D was pressed)
1 (correct passkey attempt
update_color(LOCKED)
end waitForUnlock (return 1)
end waitForUnlock (return 1)
passkey
int i = 0
char triedPin[5]
triedPin[i] = readInput()
increment i
does triedPin[i] = 'D'?
yes
end passkey (return 0)
no
does i = 4?
yes
does triedPin = '5431'?
yes
end passkey (return 1)
no
i = 0
no

System Architecture Diagram

Software Architecture



Circuit Diagram