

## HW 1: Code Lock

Your assignment is to write a program to emulate a code lock for a door. A serial port will be used to program the lock's combination code and for log messages. The joystick will be used to for user code entry. Status will be indicated to the user by a light.

### User interface:

The system will be presented to the lock operator as three primary states, locked with no user entry, locked partial user code entry, and unlocked.

When the system is armed and locked with no user code entry, the LED should be "ON". When the user begins to enter a code, the LED should blink at a rate of 2 Hertz. When the user finally presses the center button to complete a code, the lock will go to one of two states depending on the code entered: If the code matched the code programmed when the code entry started, the system should enter the unlocked state for 5 seconds with the LED "OFF" and then automatically reenter the locked/armed state. If the code entered was wrong, the system should immediately enter the locked state.

### Serial Code Programming and Log Interface:

The system will accept new code programming through the serial port only when the system is in the locked state with no user input initiated and will not accept any user input during a programming sequence.

The protocol for the programming interface is as follows. A new code programming is initiated by a character '^' being sent to the microcontroller. Following that character will be some number of characters drawn from the set {'U','D','L','R'}. Finally, a termination character 'C' will be sent to the microcontroller to end the sequence. If any invalid character is sent, the microcontroller will terminate the programming sequence until another '^' is received.

Messages should be sent from the microcontroller through the serial interface on the following events:

- '^' received, starting the code programming : "PCBG" (processor code begin)
- Incorrect character received through serial port after a '^' and before a 'C', "PCIC" (processor code incorrect character)
- New code programmed, your message must report the code "PCSC:" and report that code ending in a C (processor code store code)
- When programming code exceeds a maximum allowed length, programming is terminated "PCTL" (processor code too long)
- User code entry started. "UCBG" (user code begin)
- When user presses C, report if the code entered was correct "UCMT" (user code match) or not "UCNM " (user code not match) and if the door is unlocked "DRUL" (door unlocked)
- User code entry too long "UCTL" (user code too long)
- System automatic relock "DRLK" (door locked)

## Other notes:

- our microcontroller uses the include file: `.include "m169pdef.inc"`
- when testing, set the serial program to 4800 Baud and 2 stop bits
- The microcontroller clock speed should be set to 8 Hz

Extra credit may be given for creating various "beeps" to confirm button presses, center button press, successful matching code, unmatched code, etc....

I suggest practicing a state machine approach.

- INITIALIZATION (IO directions, UART, etc...)
- MAIN\_LOOP:
  - CHECK FOR INPUT FROM USER AND UART
  - BASED ON STATE AND NEW INPUT, DECIDE ON ACTIONS AND NEXT STATE
  - PROCESS ANY ACTIONS LIKE CHANGING LED STATE OR SENDING A CHARACTER, AND/OR PERFORMING ANY ADDITIONAL LOOP DELAY

## Final Report:

- Begin by clearly summarizing the completeness of the system. Report what testing procedures you performed. The grader has a difficult job evaluating everyone hardware design, since there is a tendency for each design to be slightly different (like using the wrong baud rate, first button press not recognized, etc...). You may only receive credit for what you report as finished and tested.
- Document design decisions and additional design specifications/details identified and implemented.
- Provide a usage manual (make it clear what the system actually does, not only what was intended)

## Initial Report:

- Identify ambiguities or desired clarification in design specification, anywhere where the provided specification is weak
- Estimate achievable specifications.
- Draft a specification document – design document

## Major tasks to be evaluated

- Capture and processing of serial input
- Storage of programmed code
- Capture and identification of user button presses
- Comparison of button presses to stored code
- LED status indicator
- Log messages
- Overall proper system operation
- Initial report
- Final Report (clarity and presentation, completeness)
- Code commenting
- Extra credit