

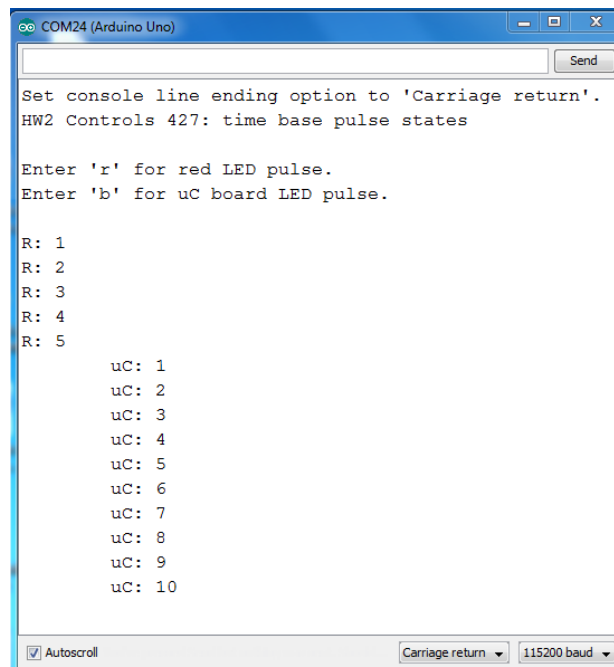
Due Thursday 9/8 before Lab. Demonstrate operation to instructor and submit zipped code to drop box.

Problem 1: State Machine creation

Convert the time HW2_TIME_BASE_pulse example from a flat architecture to a more modular version by creating two state machines to control the onboard uC LED and an external red LED. Assume the Arduino Uno or Nano pinout and atMega328 processor (A-star be gone!)

Procedure:

1. Run the HW2_TIME_BASE_pulse sketch and observe the mechanisms of LED on/off state control based on defined time thresholds.
2. Run the HW2_TIME_BASE_pulse_states_ASSIGNMENT code and note the use of Boolean flags to trigger active behaviors in the stub functions pulseRedSM() and pulseMicroSM().
3. Use the state machine in HW2_STATE_MACHINE_digital_pulse_ASSIGNMENT as a starting point to create a machine with to implement full pulseRedSM() functionality. Identify red ticks in the style shown in the console window snapshot below. (Serial.print() statements in appropriate states). Note that the ticks variable is not global, but declared and used as static inside the state machine.
4. Copy paste and edit your pulseRedSM into the pulseMicro_SM tab to complete the design. Modify the print statements to observe the tabbed spacing shown in the console snapshot. The example shows the result of entering 'r', followed by 'b' upon 'r' result completion.



```
COM24 (Arduino Uno)
Set console line ending option to 'Carriage return'.
HW2 Controls 427: time base pulse states

Enter 'r' for red LED pulse.
Enter 'b' for uC board LED pulse.

R: 1
R: 2
R: 3
R: 4
R: 5
    uC: 1
    uC: 2
    uC: 3
    uC: 4
    uC: 5
    uC: 6
    uC: 7
    uC: 8
    uC: 9
    uC: 10

Autoscroll Carriage return 115200 baud
```

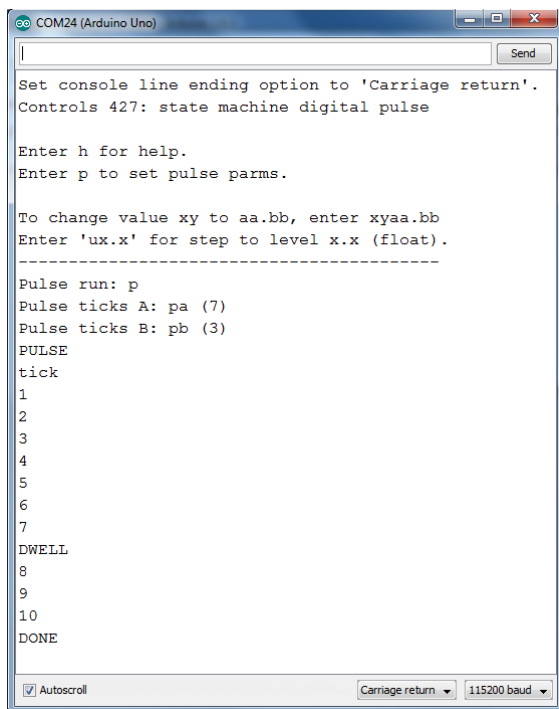
Problem 2: Hardware platform mapping

The sketch PWM_STEPS_A-star_ASSIGNMENT controls an external red LED using Timer-Counter3 of a 32U4 uC operating in 9 bit mode. Convert the design to 8 bit Timer-Counter2 operation in an Uno/Nano design. You will need to map the Timer setup as well as the loop() command levels to be consistent with an 8 bit cycle. (Starting code will only compile for an Arduino IDE board selection of A-Star 32U4.)

Problem 3. State Machine sequencing

Reconfigure the pulseDigitalLED() state machine so that the order of DWELL and PULSE is reversed in the output. Turn the external red LED on during PULSE and turn the uC LED on during DWELL.

STATE_MACHINE_digital_pulse_ASSIGNMENT



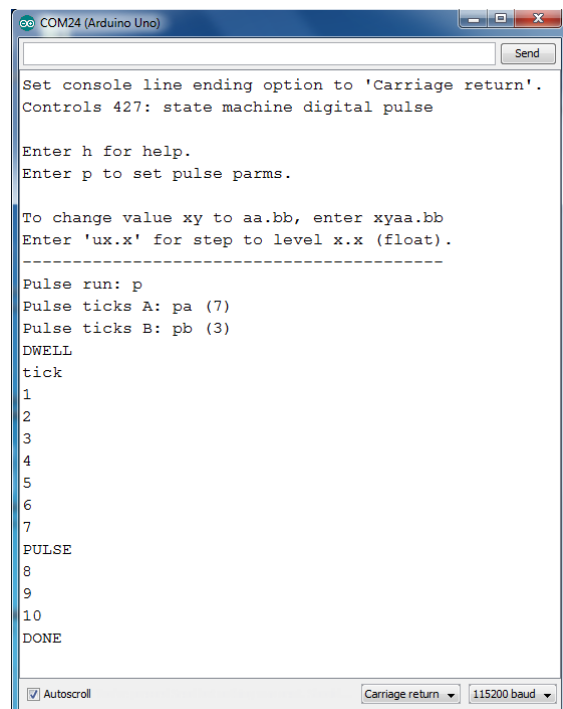
```

Set console line ending option to 'Carriage return'.
Controls 427: state machine digital pulse

Enter h for help.
Enter p to set pulse parms.

To change value xy to aa.bb, enter xyaa.bb
Enter 'ux.x' for step to level x.x (float).
-----
Pulse run: p
Pulse ticks A: pa (7)
Pulse ticks B: pb (3)
PULSE
tick
1
2
3
4
5
6
7
DWELL
8
9
10
DONE
  
```

STATE_MACHINE_digital_pulse_SOLUTION



```

Set console line ending option to 'Carriage return'.
Controls 427: state machine digital pulse

Enter h for help.
Enter p to set pulse parms.

To change value xy to aa.bb, enter xyaa.bb
Enter 'ux.x' for step to level x.x (float).
-----
Pulse run: p
Pulse ticks A: pa (7)
Pulse ticks B: pb (3)
DWELL
tick
1
2
3
4
5
6
7
PULSE
8
9
10
DONE
  
```