ECE 476/676 - Homework #7

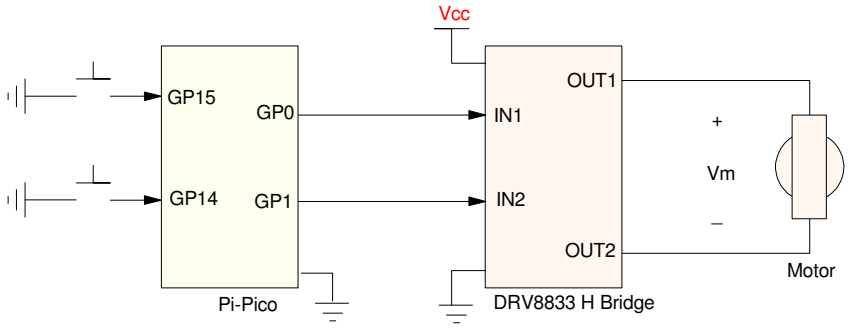
Edge and Timer Interrupts - Due Monday, October 20th

DC Motor

1) Connect the DC servo motor to your PIC board ising an H-bridge. Verify that your PIC can turn the motor on and off

- Vcc = 3.3V
- Forward: IN1: IN2 = 1 : 0
- Stop: IN1 : IN2 = 0 : 0
- Reverse: IN1 : IN2 = 0 : 1

Note: This probably doesn't work with Vcc = +5.0V. The in-rush current is too large when you apply a step input from 0% to +100% all at once.

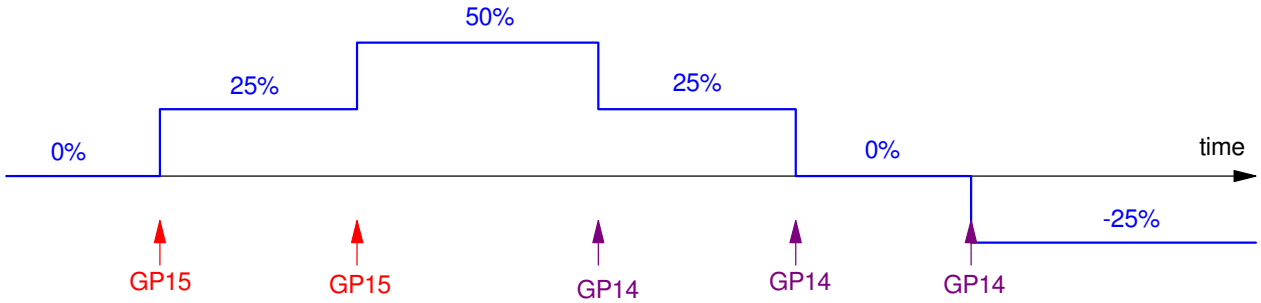


Edge Interrupts

2) Write a Python program using interrupts to allow you to turn the motor on and off:

- The speed of the motor can be adjusted from -100% to +100% of full speed
- PWM sets the speed of the motor
- Edge interrupts on GP15 and GP14 allow you to adjust the speed
  - Each press of GP15 increases the speed by +25% using a falling-edge interrupt
  - Each press of GP14 decreases the speed by -25% using a falling-edge interrupt
  - Speed clips at -100% and +100%
- The current speed is displayed on the LCD display

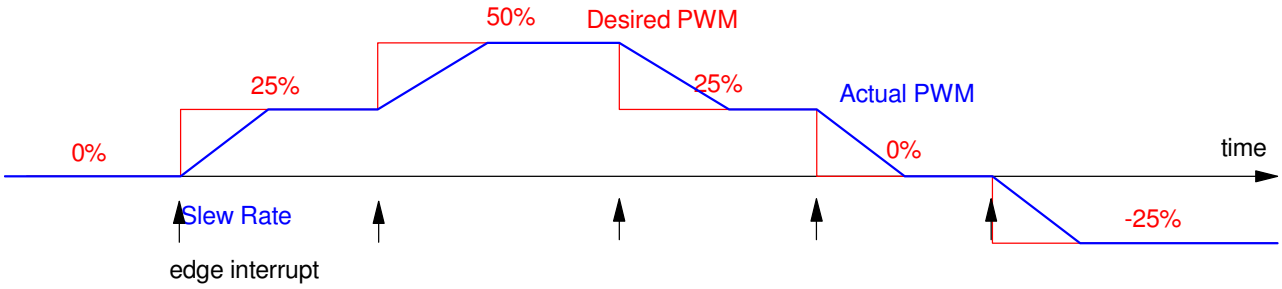
Note: 5.0V might not work due to in-rush current for the motor. If this is the case, set Vcc to +3.3V



Periodic Timer Interrupts

3) Modify this program to use timer interrupts so to add a slew-rate limit to the PWM signals. This should reduce the in-rush current to the motor, allowing the H-bridge to operate at 5.0V

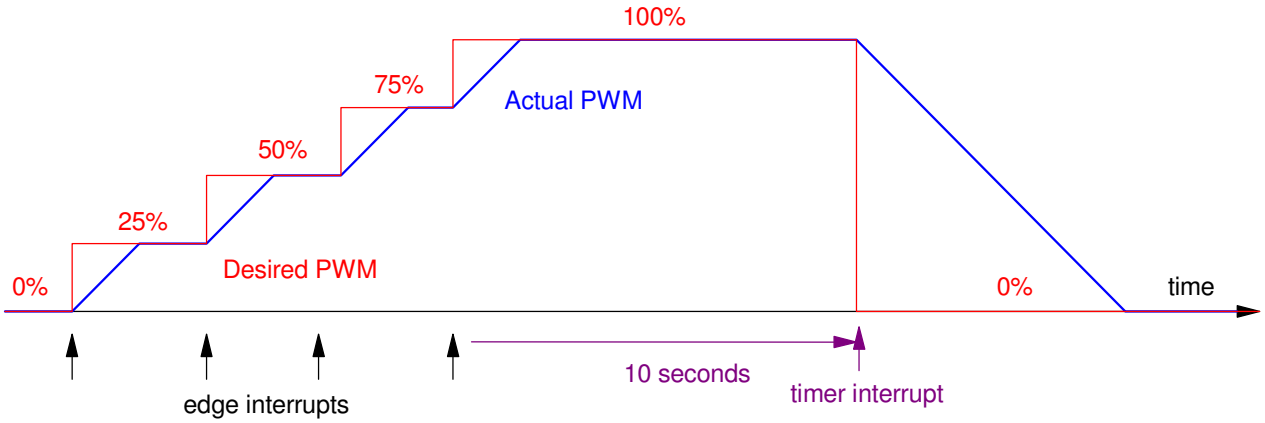
- Turn on a periodic timer-interrupt set to 10ms
- Each interrupt, the PWM changes by +0.25% or -0.25% until you reach the desired PWM (meaning it takes 1.0 second to increase the PWM by 25%. You can change the interrupt frequency if you want as long as it takes 1.0 second to ramp up or down by 25%.)
- The LCD display shows the desired PWM (from problem #1) and the actual PWM (ramping up or down from problem #2)
  - Text or graphics are OK for indicating this



One-Shot Timer Interrupts

4) Modify this program by adding a one-shot timer interrupt that turns off the motor (PWM ramps to 0%) if no buttons are pressed in 10 seconds.

- The LCD display also shows the time until the motor turns off
  - Text or graphics are OK for indicating this



Demonstration

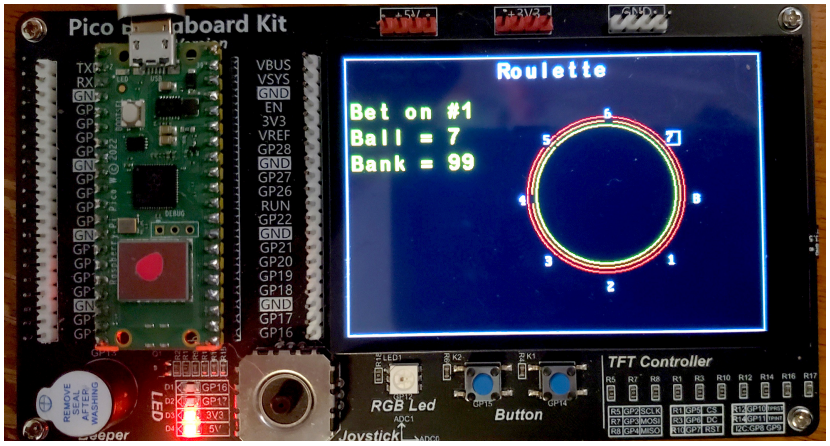
5) Demonstrate you motor controller

- In-person or
- With a video

Build the program step by step:

Step 1: Display your bet, the ball position, and your bank balance (get the display working first so you can see what's happening)

```
from time import sleep
from random import randrange
from math import sin, cos, pi
import LCD
from machine import ADC, Pin, Timer

Bet = Ball = 1
Flag = N = 0
Bank = 100

def Display():
    LCD.Circle(300, 160, 90, Red)
    LCD.Circle(300, 160, 85, Orange)
    LCD.Circle(300, 160, 80, Yellow)
    for i in range(1,9):
        x = 300 + 100*cos(i*pi/4)
        y = 160 + 100*sin(i*pi/4)
        LCD.Text(str(i), x, y, White, Black)

LCD.Init()
Yellow = LCD.RGB(250,250,0)
White =  LCD.RGB(250,250,250)
Black =  LCD.RGB(0,0,0)
Pink = LCD.RGB(250,100,100)
Red = LCD.RGB(250,0,0)
Orange = LCD.RGB(250,150,0)

LCD.Clear(Black)
LCD.Box(0,0,479,319, White)
LCD.Title('Roulette', White, Black)
Display()

while(1):

    msg = 'Bet on #' + str(Bet)
    LCD.Text2(msg, 5, 50, Yellow, Black)
    msg = 'Ball = ' + str(Ball)
    LCD.Text2(msg, 5, 80, Yellow, Black)
    msg = 'Bank = ' + str(Bank) + '  '
    LCD.Text2(msg, 5, 110, Yellow, Black)

    sleep(0.1)
```



Step 2: Get the analog input to work.

- Pushing the joystick up, increses the number you're betting on
- Pushing the joystick down decreases the number you're betting on

Check that the display changes when you press the joystick

```
x0 = a2d1.read_u16()

while(1):
    dx = (a2d1.read_u16() - x0)
    if(dx > 10_000):
        Bet = min(Bet+1, 8)
    if(dx < -10_000):
        Bet = max(Bet-1, 1)

    msg = 'Bet on #' + str(Bet)
    LCD.Text2(msg, 5, 50, Yellow, Black)
    msg = 'Ball = ' + str(Ball)
    LCD.Text2(msg, 5, 80, Yellow, Black)
    msg = 'Bank = ' + str(Bank) + '  '
    LCD.Text2(msg, 5, 110, Yellow, Black)

    sleep(0.1)
```

Check:

- Number you bet on can be inceased or decrased as you push the joystick up and down

Step 3: Start the game and generate a random number on a falling-edge interrupt on GP15

```
Bet = Ball = 1
Flag = N = 0
Bank = 100

pin = Pin(15,Pin.IN,Pin.PULL_UP)
LED = Pin(16, Pin.OUT)
Beeper = Pin(13, Pin.OUT)

def IntServe(pin):
    global Ball
    global Flag
    global Bet
    global N
    Flag = 1
    X = randrange(9)
    if(X == 0):
        X = X - 1
    if(X == 0):
        X = 8
    N = 40 + X - Ball
    print('Edge Interrupt:  N = ', N)

pin.irq(trigger=Pin.IRQ_FALLING, handler=IntServe)
```

Check:

- Pressing GP15 changes N every time it's pressed

Step 4: Turn on the timer interrupts

- Start the interrupt with the edge interrupt
- Continue for N times

```
tim = Timer()

def Wheel(pin1):
    global N
    global Ball
    x = 300 + 100*cos(Ball*pi/4)
    y = 160 + 100*sin(Ball*pi/4)
    LCD.Box(x, y, x+15, y+15, Black)
    Ball = (Ball % 8) + 1
    N = N - 1
    if(N > 0):
        tim.init(period = int(100 + 1000/N),mode=Timer.ONE_SHOT,callback=Wheel)
    LED.toggle()
    x = 300 + 100*cos(Ball*pi/4)
    y = 160 + 100*sin(Ball*pi/4)
    LCD.Box(x, y, x+15, y+15, White)
    Beep()
```

When turned on

- Flag = 1
- N > 0

the timer interrupts repeat N times then stop

Step 4: Calculate winnings

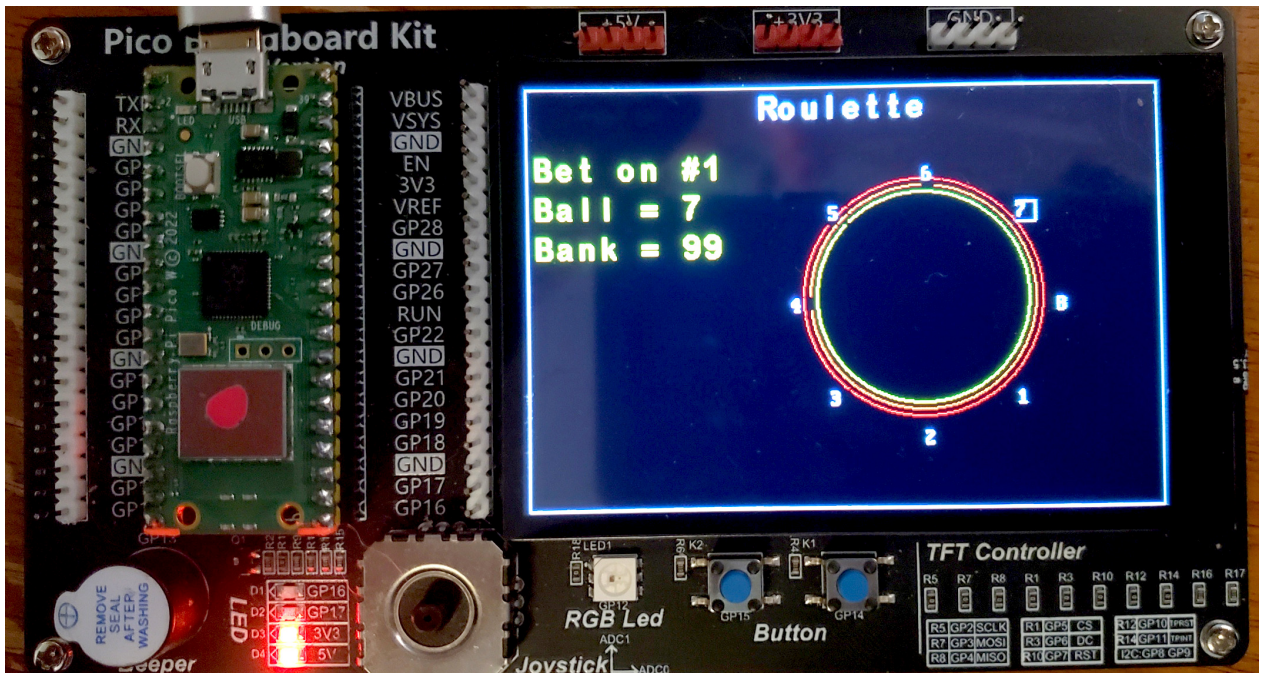
If the game just ended (Flag == 1 and N == 0), check to see if you won

```
while(1):
    dx = (a2d1.read_u16() - x0)
    if(dx > 10_000):
        Bet = min(Bet+1, 8)
    if(dx < -10_000):
        Bet = max(Bet-1, 1)

    if(Flag == 1):
        if(N == 0):
            Flag = 0
            if(Ball == Bet):
                Bank += 8
            else:
                Bank -= 1

    msg = 'Bet on #' + str(Bet)
    LCD.Text2(msg, 5, 50, Yellow, Black)
    msg = 'Ball = ' + str(Ball)
    LCD.Text2(msg, 5, 80, Yellow, Black)
    msg = 'Bank = ' + str(Bank) + '  '
    LCD.Text2(msg, 5, 110, Yellow, Black)

    sleep(0.1)
```

Final Code:

```
from time import sleep
from random import randrange
from math import sin, cos, pi
import LCD
from machine import ADC, Pin, Timer

a2d0 = ADC(0)
a2d1 = ADC(1)

Bet = Ball = 1
Flag = N = 0
Bank = 100

x0 = a2d1.read_u16()

pin = Pin(15,Pin.IN,Pin.PULL_UP)
LED = Pin(16, Pin.OUT)
Beeper = Pin(13, Pin.OUT)

# --- Edge Interrupts

def IntServe(pin):
    global Ball
    global Flag
    global Bet
    global N
    Flag = 1
    X = randrange(9)
    if(X == 0):
        X = X - 1
    if(X == 0):
        X = 8
    N = 40 + X - Ball
    tim.init(period = 200,mode=Timer.ONE_SHOT,callback=Wheel)

pin.irq(trigger=Pin.IRQ_FALLING, handler=IntServe)

# --- Timer Interrupts

tim = Timer()
N = 0

def Wheel(pin1):
    global N
    global Ball
    x = 300 + 100*cos(Ball*pi/4)
    y = 160 + 100*sin(Ball*pi/4)
    LCD.Box(x, y, x+15, y+15, Black)
    Ball = (Ball % 8) + 1
    N = N - 1
    if(N > 0):
        tim.init(period = int(100 + 1000/N),mode=Timer.ONE_SHOT,callback=Wheel)
    LED.toggle()
    x = 300 + 100*cos(Ball*pi/4)
    y = 160 + 100*sin(Ball*pi/4)
    LCD.Box(x, y, x+15, y+15, White)
    Beep()
```

```
# --- Beep for 50ms

def Beep():
    Beeper.value(1)
    sleep(0.05)
    Beeper.value(0)

# --- Display Routine

def Display():
    LCD.Circle(300, 160, 90, Red)
    LCD.Circle(300, 160, 85, Orange)
    LCD.Circle(300, 160, 80, Yellow)
    for i in range(1,9):
        x = 300 + 100*cos(i*pi/4)
        y = 160 + 100*sin(i*pi/4)
        LCD.Text(str(i), x, y, White, Black)

LCD.Init()
Yellow = LCD.RGB(250,250,0)
White =  LCD.RGB(250,250,250)
Black =  LCD.RGB(0,0,0)
Pink = LCD.RGB(250,100,100)
Red = LCD.RGB(250,0,0)
Orange = LCD.RGB(250,150,0)

LCD.Clear(Black)
LCD.Box(0,0,479,319, White)
LCD.Title('Roulette', White, Black)
Display()

# --- Main Loop

while(1):
    dx = (a2d1.read_u16() - x0)
    if(dx > 10_000):
        Bet = min(Bet+1, 8)
    if(dx < -10_000):
        Bet = max(Bet-1, 1)

    if(Flag == 1):
        if(N == 0):
            Flag = 0
            if(Ball == Bet):
                Bank += 8
            else:
                Bank -= 1

    msg = 'Bet on #' + str(Bet)
    LCD.Text2(msg, 5, 50, Yellow, Black)
    msg = 'Ball = ' + str(Ball)
    LCD.Text2(msg, 5, 80, Yellow, Black)
    msg = 'Bank = ' + str(Bank) + '  '
    LCD.Text2(msg, 5, 110, Yellow, Black)

    sleep(0.1)
```



Checks:

	Bet	Winning #	Bank
•	4	1	95
•	5	8	94
•	5	6	93
•	2	8	92
•	1	4	91
•	6	7	90
•	6	7	89
•	6	8	88
•	7	4	87
•	8	8	95
•	8	4	94

Looks OK

- Numbers look random
- Wheel is spinning
- Stops on a number
- I lose $1 when wrong
- I gain $8 when I win