ECE 476/676 - Homework #7

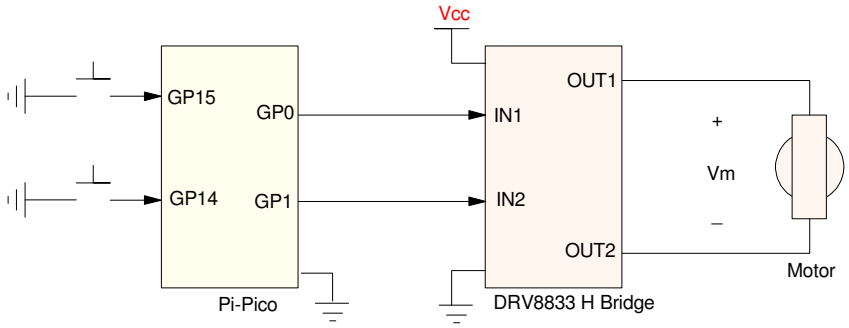
Edge and Timer Interrupts - Due Monday, October 20th

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

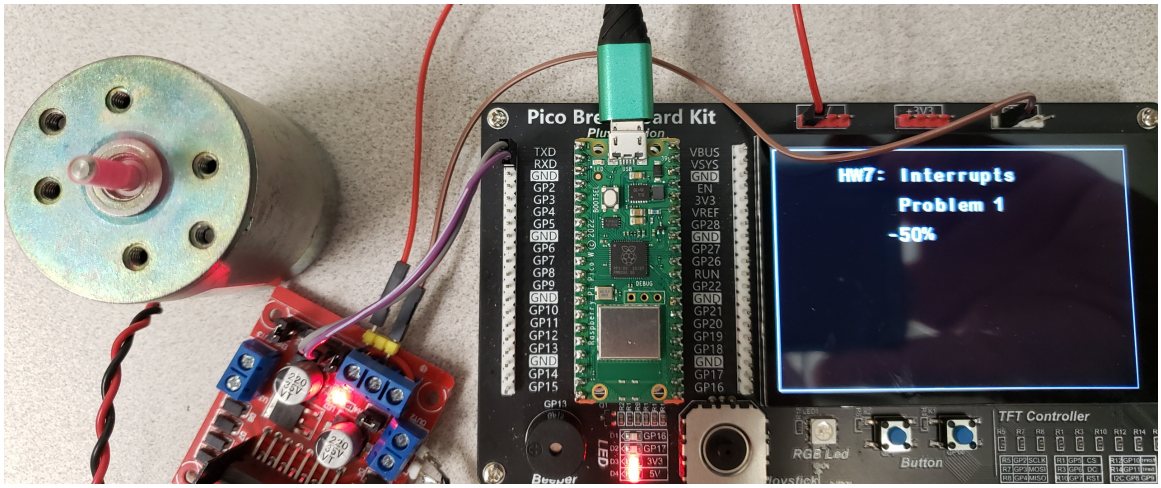
Note: This probably doesn't work with Vcc = +5.0V. The in-rush current is too large when you apply a step input from 0% to +100% all at once.



Program Description:

- The main routine counts up to 100 in steps of 25
- Then counts down to -100 in steps of -25
- And repeats

Each count, the PWM is adjusted



Python Code

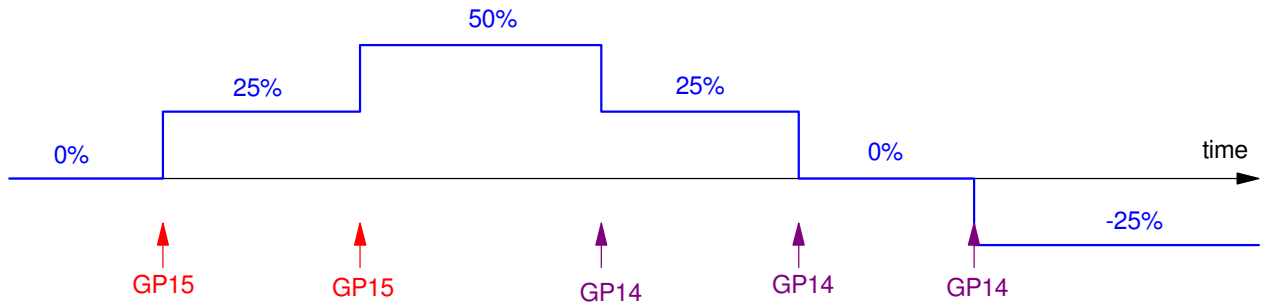
```
--- Problem 1 ---

from machine import Pin, PWM
from time import sleep_ms
import LCD_16x24 as LCD

# Subroutines
def PWM_Out(pct):
    pct = int(pct)
    pct = max(-100, min(100, pct))
    if(pct > 0):
       fwd.duty_u16(int(pct*655))
       rev.duty_u16(0)
    else:
       fwd.duty_u16(0)
       rev.duty_u16(int(abs(pct)*655))

# intialize PWM outputs
fwd = PWM(Pin(0))
fwd.freq(100)
fwd.duty_u16(0)
rev = PWM(Pin(1))
rev.freq(100)
rev.duty_u16(0)


# initialize the LCD display
LCD.Init()
Black = LCD.RGB(0,0,0)
White = LCD.RGB(250,250,250)
Grey  = LCD.RGB(100,100,100)

LCD.Clear(Black)

LCD.Box(1,1,478,318,White)
LCD.Text3('HW7: Interrupts', 240-19*8, 25, White, Black)
LCD.Text3('Problem 1', 240 - 9*8, 65, White, Black)

pct = 0
dx = 25

while(1):
    if(pct >= 100):
        dx = -abs(dx)
    if(pct <= -100):
        dx = abs(dx)
    pct = pct + dx;
    PWM_Out(pct)
    print(pct)
    LCD.Text3(str(pct) + '%  ', 50, 150, White, Black)
    sleep_ms(1000)
```

Edge Interrupts

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

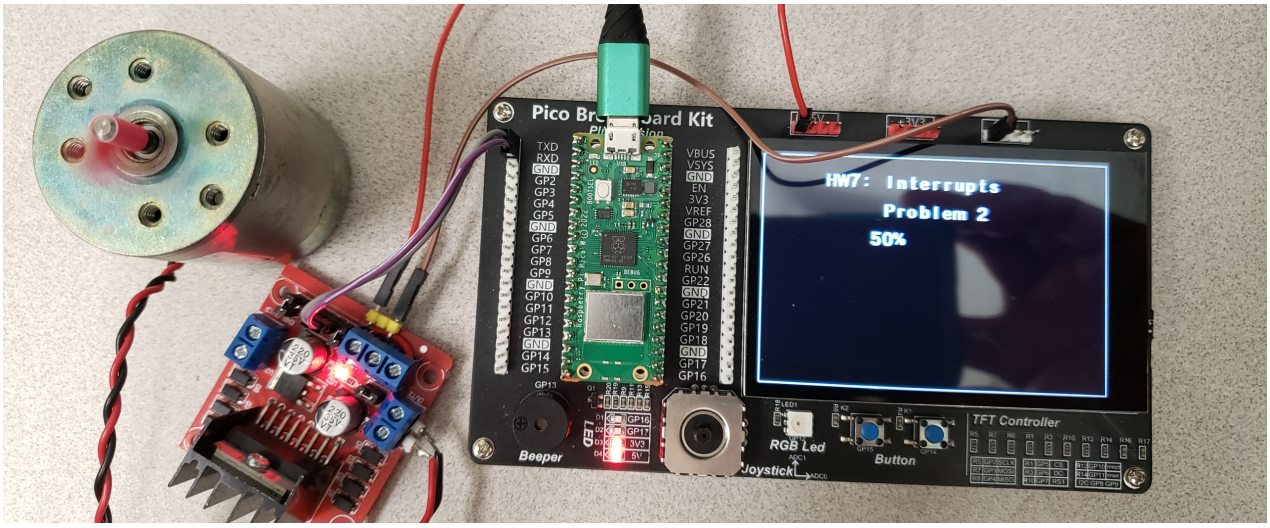
- The speed of the motor can be adjusted from -100% to +100% of full speed
- PWM sets the speed of the motor
- Edge interrupts on GP15 and GP14 allow you to adjust the speed
  - Each press of GP15 increases the speed by +25% using a falling-edge interrupt
  - Each press of GP14 decreases the speed by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



Program:

- Strip the main routine to just a display function
- Add edge interrupts to change pct



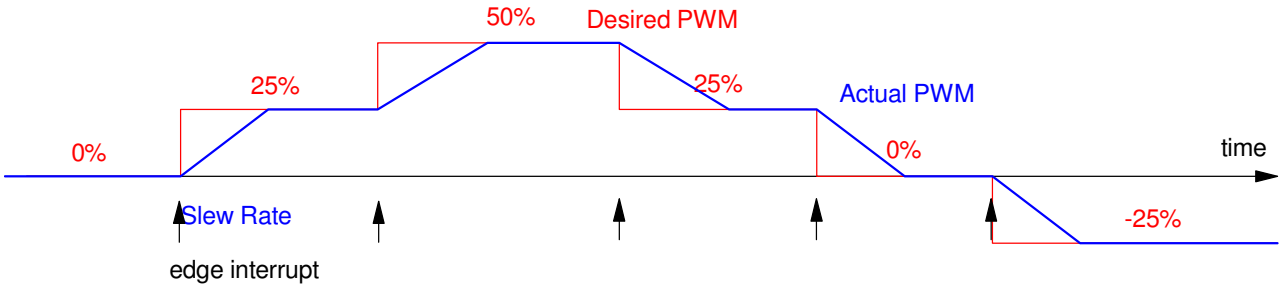
Python Code

```
from machine import Pin, PWM
from time import sleep_ms
import LCD_16x24 as LCD

# Subroutines
def PWM_Out(pct):
    pct = int(pct)
    pct = max(-100, min(100, pct))
    if(pct > 0):
       fwd.duty_u16(int(pct*655))
       rev.duty_u16(0)
    else:
       fwd.duty_u16(0)
       rev.duty_u16(int(abs(pct)*655))

# Edge Interrupts

p15 = Pin(15,Pin.IN,Pin.PULL_UP)
def Speed_Up(p15):
    global pct, flag
    if(pct < 100):
        pct += 25
    flag = 1

p15.irq(trigger=Pin.IRQ_FALLING, handler=Speed_Up)

p14 = Pin(14,Pin.IN,Pin.PULL_UP)
def Speed_Down(p14):
    global pct, flag
    if(pct > -100):
        pct -= 25
    flag = 1

p14.irq(trigger=Pin.IRQ_FALLING, handler=Speed_Down)

# intialize PWM outputs
fwd = PWM(Pin(0))
fwd.freq(100)
fwd.duty_u16(0)
rev = PWM(Pin(1))
rev.freq(100)
rev.duty_u16(0)

# initialize the LCD display
LCD.Init()
Black = LCD.RGB(0,0,0)
White = LCD.RGB(250,250,250)
LCD.Clear(Black)

LCD.Box(1,1,478,318,White)
LCD.Text3('HW7: Interrupts', 240-19*8, 25, White, Black)
LCD.Text3('Problem 2', 240 - 9*8, 65, White, Black)

pct = flag = 0
dx = 25

while(1):
    if(flag):
        flag = 0
        PWM_Out(pct)
        print(pct)
        LCD.Text3(str(pct) + '%  ', 50, 150, White, Black)
```

Periodic Timer Interrupts

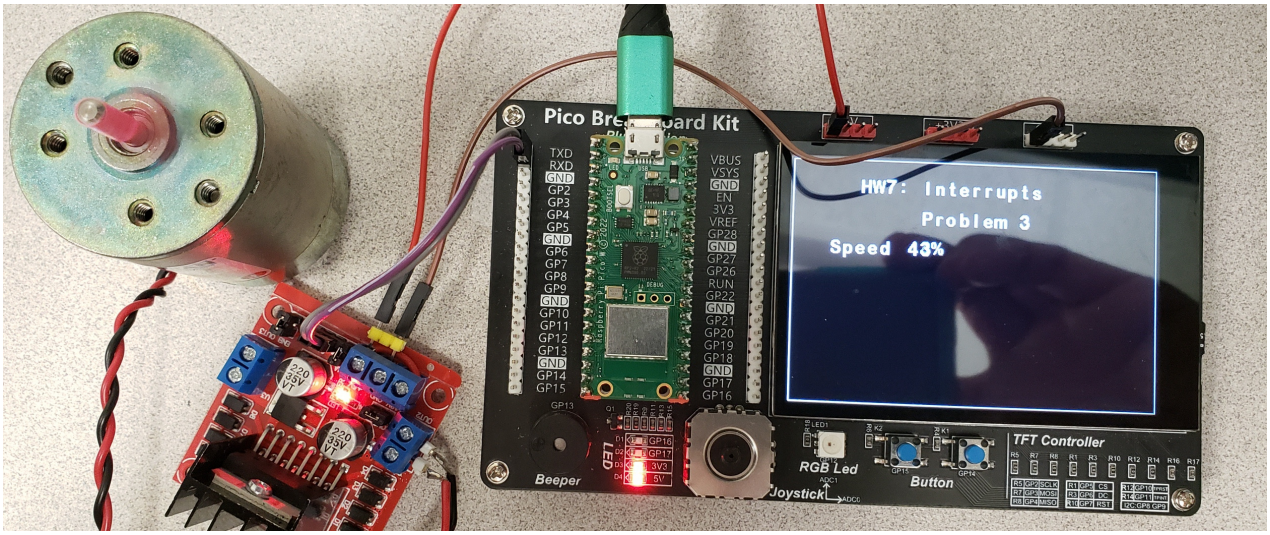
3) Modify this program to use timer interrupts so to add a slew-rate limit to the PWM signals. This should reduce the in-rush current to the motor, allowing the H-bridge to operate at 5.0V

- 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



Change the previous code:

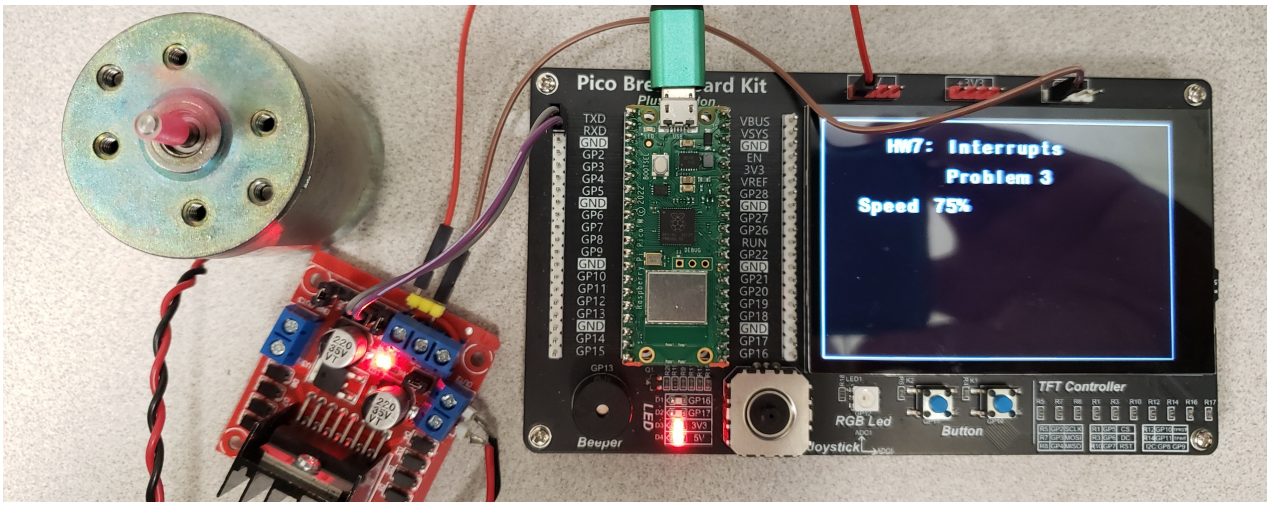
- Edge interrupts change a global variable, Ref
- Timer interrupts are set for 100ms
- Each timer interrupt, you count up or down with pct, stopping when pct = Ref



Python Code

```
from machine import Pin, PWM, Timer
from time import sleep_ms
import LCD_16x24 as LCD

# Global Variables
pct = flag = Ref = 0
dx = 25

# Subroutines
def PWM_Out(pct):
    pct = int(pct)
    pct = max(-100, min(100, pct))
    if(pct > 0):
       fwd.duty_u16(int(pct*655))
       rev.duty_u16(0)
    else:
       fwd.duty_u16(0)
       rev.duty_u16(int(abs(pct)*655))

# Edge Interrupts

p15 = Pin(15,Pin.IN,Pin.PULL_UP)
def Speed_Up(p15):
    global Ref
    if(Ref < 100):
        Ref += 25

p15.irq(trigger=Pin.IRQ_FALLING, handler=Speed_Up)

p14 = Pin(14,Pin.IN,Pin.PULL_UP)
def Speed_Down(p14):
    global Ref
    if(Ref > -100):
        Ref -= 25

p14.irq(trigger=Pin.IRQ_FALLING, handler=Speed_Down)

# Timer Interrupts

# 100Hz for 10 seconds
def tick(timer):
    global Ref, pct, flag
    if(pct < Ref):
        pct += 1
        flag = 1
    if(pct > Ref):
        pct -= 1
        flag = 1

tim = Timer()
tim.init(period = 100, mode=Timer.PERIODIC, callback=tick)


# intialize PWM outputs
fwd = PWM(Pin(0))
fwd.freq(100)
fwd.duty_u16(0)
rev = PWM(Pin(1))
rev.freq(100)
rev.duty_u16(0)

# initialize the LCD display
LCD.Init()
```

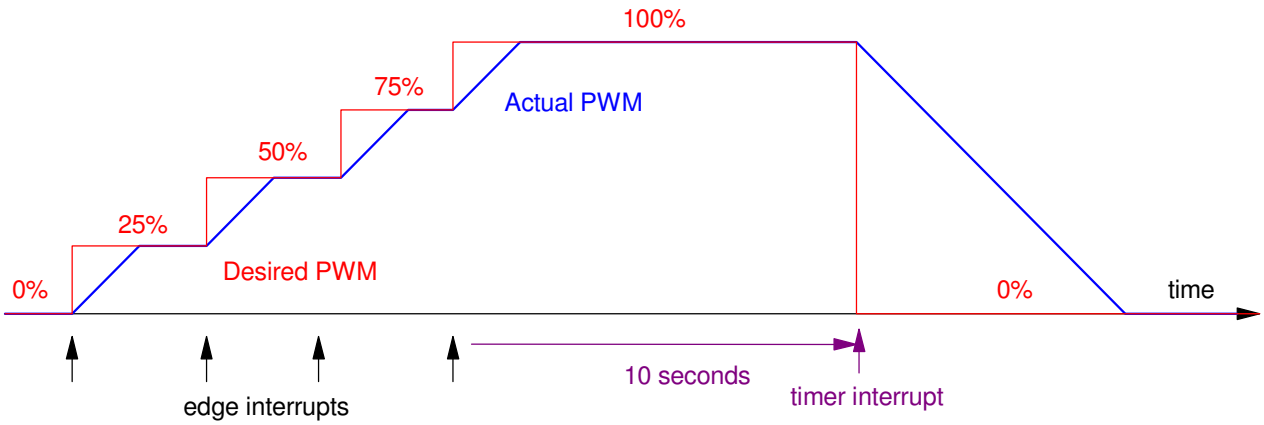
```
Black = LCD.RGB(0,0,0)
White = LCD.RGB(250,250,250)
Grey  = LCD.RGB(100,100,100)

LCD.Clear(Black)

LCD.Box(1,1,478,318,White)
LCD.Text3('HW7: Interrupts', 240-19*8, 25, White, Black)
LCD.Text3('Problem 4', 240 - 9*8, 65, White, Black)
LCD.Text3('Speed', 50, 105, White, Black)

while(1):
    if(flag):
        flag = 0
        PWM_Out(pct)
        LCD.Text3(str(pct) + '%  ', 150, 105, White, Black)
        print(Ref, pct)
```



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



Change in the previous code

- Each time you press a button, a one-shot interrupt is triggered 10 seconds in the future
- Each button press resets the one-shot interrupt
- Once 10 seconds elapses, Ref is reset to 0

Code

```
from machine import Pin, PWM, Timer
from time import sleep_ms
import LCD_16x24 as LCD

# Global Variables
pct = flag = Ref = 0
dx = 25

# Subroutines
def PWM_Out(pct):
    pct = int(pct)
    pct = max(-100, min(100, pct))
    if(pct > 0):
       fwd.duty_u16(int(pct*655))
       rev.duty_u16(0)
    else:
       fwd.duty_u16(0)
       rev.duty_u16(int(abs(pct)*655))

# Edge Interrupts

p15 = Pin(15,Pin.IN,Pin.PULL_UP)
def Speed_Up(p15):
    global Ref
    if(Ref < 100):
        Ref += 25
    t_stop.init(period = 10000, mode=Timer.ONE_SHOT, callback=Stop)

p15.irq(trigger=Pin.IRQ_FALLING, handler=Speed_Up)

p14 = Pin(14,Pin.IN,Pin.PULL_UP)
def Speed_Down(p14):
    global Ref
    if(Ref > -100):
        Ref -= 25
    t_stop.init(period = 10000, mode=Timer.ONE_SHOT, callback=Stop)

p14.irq(trigger=Pin.IRQ_FALLING, handler=Speed_Down)

# Timer Interrupts

# 100Hz for 10 seconds
def tick(timer):
    global Ref, pct, flag
    if(pct < Ref):
        pct += 1
        flag = 1
    if(pct > Ref):
        pct -= 1
        flag = 1

tim = Timer()
tim.init(period = 100, mode=Timer.PERIODIC, callback=tick)

t_stop = Timer()
def Stop(timer):
    global Ref
    Ref = 0


# intialize PWM outputs
fwd = PWM(Pin(0))
fwd.freq(100)
fwd.duty_u16(0)
```

```
rev = PWM(Pin(1))
rev.freq(100)
rev.duty_u16(0)

# initialize the LCD display
LCD.Init()
Black = LCD.RGB(0,0,0)
White = LCD.RGB(250,250,250)
Grey  = LCD.RGB(100,100,100)

LCD.Clear(Black)

LCD.Box(1,1,478,318,White)
LCD.Text3('HW7: Interrupts', 240-19*8, 25, White, Black)
LCD.Text3('Problem 3', 240 - 9*8, 65, White, Black)
LCD.Text3('Speed', 50, 105, White, Black)

while(1):
    if(flag):
        flag = 0
        PWM_Out(pct)
        LCD.Text3(str(pct) + '%  ', 150, 105, White, Black)
        print(Ref, pct)
```

Demonstration

5) Demonstrate you motor controller
   - In-person or
   - With a video