

hydra

```
import RPi.GPIO as gpio
import time
import sys
import termios
import tty
```

```
def init():
    gpio.setmode(gpio.BOARD)
    gpio.setup(13, gpio.OUT)
    gpio.setup(15, gpio.OUT)
    gpio.setup(18, gpio.OUT)
    gpio.setup(22, gpio.OUT)
```

#initiating pins for bot movement

```
def init_arm():
    gpio.setmode(gpio.BOARD)
    gpio.setup(37, gpio.OUT)
```

#initiating pins for grip movement

```
def init_crane():
    gpio.setmode(gpio.BOARD)
    gpio.setup(38, gpio.OUT)
    gpio.setup(40, gpio.OUT)
```

#initiating pins for crane movement

```
def forward(tf):
    init()
    gpio.output(13, 0)
    gpio.output(15, 1)
    gpio.output(18, 0)
    gpio.output(22, 1)
    time.sleep(tf)
    gpio.cleanup()
```

#forward function

```
def reverse(tf):
    init()
    gpio.output(13, 0)
    gpio.output(15, 1)
    gpio.output(18, 1)
    gpio.output(22, 0)
    time.sleep(tf)
    gpio.cleanup()
```

#reverse function

```
def turn_right(tf):
    #turn_right function
```

hydra

```
init()
gpio.output(13, 0)
gpio.output(15, 1)
gpio.output(18, 0)
gpio.output(22, 0)
time.sleep(tf)
gpio.cleanup()
```

```
def turn_left(tf):                                #turn_left function
```

```
    init()
    gpio.output(13, 0)
    gpio.output(15, 0)
    gpio.output(18, 0)
    gpio.output(22, 1)
    time.sleep(tf)
    gpio.cleanup()
```

```
def pivot_right(tf):                             #pivot right function
```

```
    init()
    gpio.output(13, 0)
    gpio.output(15, 1)
    gpio.output(18, 1)
    gpio.output(22, 0)
    time.sleep(tf)
    gpio.cleanup()
```

```
def pivot_left(tf):                             #pivot left function
```

```
    init()
    gpio.output(13, 1)
    gpio.output(15, 0)
    gpio.output(18, 0)
    gpio.output(22, 1)
    time.sleep(tf)
    gpio.cleanup()
```

```
def grip(tf):                                    #grip function for gripper
```

```
    init_arm()
    frequency_hertz = 50
    pwm = GPIO.PWM(37, frequency_hertz)

    left_position = 0.40
    right_position = 2.5
```

```

                                hydra
ms_per_cycle = 1000 / frequency_hertz

duty_cycle_percentage = 0.40 * 100 / ms_per_cycle

pwm.start(duty_cycle_percentage)
time.sleep(.5)

pwm.stop()
gpio.cleanup()

def loose(tf):                                #loose function for gripper

    init_arm()
    frequency_hertz = 50
    pwm = GPIO.PWM(37, frequency_hertz)

    left_position = 0.40
    right_position = 2.5

    ms_per_cycle = 1000 / frequency_hertz

    duty_cycle_percentage = 2.5 * 100 / ms_per_cycle

    pwm.start(duty_cycle_percentage)
    time.sleep(.5)

    pwm.stop()
    gpio.cleanup()

def up(tf):                                #Upward function for crane

    init_crane()
    gpio.output(38, 0)
    gpio.output(40, 1)
    time.sleep(tf)
    gpio.cleanup()

def down(tf):                                #Downward function for crane

    init_crane()
    gpio.output(38, 1)
    gpio.output(40, 0)
    time.sleep(tf)
    gpio.cleanup()

```

hydra

```
def getch():                                #key binding

    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    try:
        tty.setraw(sys.stdin.fileno())
        ch = sys.stdin.read(1)
    finally:
        termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch

while True:
    # Keyboard character retrieval method is called and saved
    # into variable
    char = getch()

    # The car will drive forward when the "w" key is pressed
    if(char == "w"):
        forward(0.5)

    # The car will reverse when the "s" key is pressed
    if(char == "s"):
        reverse(0.5)

    # The "a" key will toggle the steering left
    if(char == "a"):
        turn_left(0.2)

    # The "d" key will toggle the steering right
    if(char == "d"):
        turn_right(0.2)

    # The "q" key will pivot left
    if(char == "q"):
        pivot_left(0.2)

    # The "e" key will pivot right
    if(char == "e"):
        pivot_right(0.2)

    # The "u" key will move up crane
    if(char == "u"):
        up(0.1)

    # The "i" key will move down crane
    if(char == "i"):
        down(0.1)

    # The "g" key will grip
```

hydra

```
if(char == "g"):
    grip(1)
```

```
# The "h" key will loose
if(char == "h"):
    loose(1)
```

```
# The "x" key will break the loop and exit the program
if(char == "x"):
    print("Program Ended")
    break
```

```
# The keyboard character variable will be set to blank, ready
# to save the next key that is pressed
char = ""
```

```
gpio.cleanup()
```

```
#forward(2)
#brakes(1)
#reverse(3)
#brakes(1)
#turn_right(2)
#brakes(1)
#turn_left(2)
#brakes(1)
#pivot_right(2)
#brakes(1)
#pivot_left(2)
```