```
#-----------------------------------------------

# Python program to interact with and
# control a MakeBlock Robot.
# Author: David Cotterill-Drew 2016
# Email: roboshopz@gmail.com
#-----------------------------------------------

# Imports
#-----------------------------------------------


from bluepy.btle import *
from Tkinter import *
import Tkinter as tk
from tkFileDialog   import askopenfilename
from PIL import Image, ImageTk

root = tk.Tk()

#-----------------------------------------------
# Global variables
#-----------------------------------------------

data2=0
conn = Peripheral("00:0D:19:12:3B:38")
dataflag=0
#-----------------------------------------------
# Speed & Range Frames
#-----------------------------------------------

framespeed = Frame(root,width=600,height=1000)
framespeed.grid(row=1, column=1, padx=10, pady=10)

framerange = Frame(root,width=600,height=1000)
framerange.grid(row=0, column=1, padx=10, pady=10)

frameleds = Frame(root,width=600,height=1000)
frameleds.grid(row=1, column=2, padx=10, pady=10)

frameservo = Frame(root,width=600,height=1000)
frameservo.grid(row=0, column=2, padx=10, pady=10)
```

```
#-----------------------------------------------
# Range Data & Label
#-----------------------------------------------

dattext = Label(framerange, text="RANGE cms",fg='bl
ack')
dattext.grid(row=0, column=0, padx=10, pady=5)

dat = Label(framerange, text=data2,fg='red',bg='whi
te')
dat.grid(row=2, column=0, padx=0, pady=0)

#-----------------------------------------------
# Handle 0x000e notification
#-----------------------------------------------

class NotificationHandler(DefaultDelegate):

    def handleNotification(self, cHandle, data):

            data2=ord(data)

         dat.configure(text=data2)

#-----------------------------------------------
# Speed Slider
#-----------------------------------------------

speed = IntVar()
speed = Scale(framespeed, from_=1, to=9,orient=HORI
ZONTAL, variable=speed, fg='red')
speed.grid(row=1, column=0, padx=10, pady=0)

#-----------------------------------------------
# Direction & Arm/Grip Frames
#-----------------------------------------------

frame = Frame(root,width=600,height=1000)
frame.grid(row=0, column=0, padx=10, pady=10)
framearm = Frame(root,width=600,height=1000)
framearm.grid(row=1, column=0, padx=10, pady=10)
```

```
#-----------------------------------------------
# More Labels
#-----------------------------------------------

nav= Label(frame, text="DIRECTION",fg='black')
nav.grid(row=0, column=1, padx=0, pady=10)
arm= Label(framearm, text="ARM",fg='black')
arm.grid(row=0, column=1, padx=0, pady=0)
grip= Label(framearm, text="GRIP",fg='black')
grip.grid(row=0, column=0, padx=0, pady=0)
speed1= Label(framespeed, text="SPEED",fg='black')
speed1.grid(row=0, column=0, padx=0, pady=0)

#-----------------------------------------------
# Title
#-----------------------------------------------

root.wm_title("Bluetooth MakeBlock")

#-----------------------------------------------
# Photo's
#-----------------------------------------------

photo1 = tk.PhotoImage(file="rsz_left.png")
photo2 = tk.PhotoImage(file="rsz_right.png")
photo3 = tk.PhotoImage(file="rsz_up.png")
photo4 = tk.PhotoImage(file="rsz_down.png")
photo5 = tk.PhotoImage(file="rsz_stop.png")
photo6 = tk.PhotoImage(file="enter.png")

#-----------------------------------------------
# Menu
#-----------------------------------------------

menu = Menu(root)

root.config(menu=menu)
filemenu = Menu(menu)
helpmenu = Menu(menu)

def About():
    print "MakeBlock Bluetooth Robot Control"
```

```
def Settings():
    print "will write it soon"

filemenu.add_command(label="Exit", command=root.qui
t)
menu.add_cascade(label="Help", menu=helpmenu)
helpmenu.add_command(label="About...", command=Abou
t)
helpmenu.add_command(label="Settings...", command=S
ettings)

#---------------------------------------------
# Control Functions
#---------------------------------------------

def forward():
   conn.writeCharacteristic(0x0011, '\x66', False)
   conn.disconnect

def back():
   conn.writeCharacteristic(0x0011, '\x62', False)
   conn.disconnect

def right():
   conn.writeCharacteristic(0x0011, '\x72', False)
   conn.disconnect

def left():
   conn.writeCharacteristic(0x0011, '\x6C', False)
   conn.disconnect

def stop():
   conn.writeCharacteristic(0x0011, '\x73', False)
   conn.disconnect

def Holder_Up():
   conn.writeCharacteristic(0x0011, '\x61', False)
   conn.disconnect

def Holder_Down():
   conn.writeCharacteristic(0x0011, '\x63', False)
```

```
   conn.disconnect

def Hand_open():
   conn.writeCharacteristic(0x0011, '\x64', False)
   conn.disconnect

def Hand_close():
   conn.writeCharacteristic(0x0011, '\x65', False)
   conn.disconnect

def speed_1():
   conn.writeCharacteristic(0x0011, '\x31', False)
   conn.disconnect

def speed_2():
   conn.writeCharacteristic(0x0011, '\x32', False)
   conn.disconnect

def speed_3():
   conn.writeCharacteristic(0x0011, '\x33', False)
   conn.disconnect

def speed_4():
   conn.writeCharacteristic(0x0011, '\x34', False)
   conn.disconnect

def speed_5():
   conn.writeCharacteristic(0x0011, '\x35', False)
   conn.disconnect

def speed_6():
   conn.writeCharacteristic(0x0011, '\x36', False)
   conn.disconnect

def speed_7():
   conn.writeCharacteristic(0x0011, '\x37', False)
   conn.disconnect

def speed_8():
   conn.writeCharacteristic(0x0011, '\x38', False)
   conn.disconnect
```

```
def speed_9():
   conn.writeCharacteristic(0x0011, '\x39', False)
   conn.disconnect

def RangeOn():
   conn.setDelegate(NotificationHandler())
   conn.writeCharacteristic(0x0011, '\x6d', False)

   conn.waitForNotifications(1)
   conn.disconnect

def RangeOff():
    conn.writeCharacteristic(0x0011, '\x6e', False)

    dat.configure(text="Off")
    conn.disconnect

def ledsoff():
   conn.writeCharacteristic(0x0011, '\x67', False)
   conn.disconnect

def ledson():
   conn.writeCharacteristic(0x0011, '\x68', False)
   conn.disconnect

def servo90():
   conn.writeCharacteristic(0x0011, '\x69', False)
   conn.disconnect

def servo0():
   conn.writeCharacteristic(0x0011, '\x6a', False)
   conn.disconnect

def servo180():
   conn.writeCharacteristic(0x0011, '\x6b', False)
   conn.disconnect

def ledsblue():
   conn.writeCharacteristic(0x0011, '\x78', False)
   conn.disconnect

def ledsred():
```

```
   conn.writeCharacteristic(0x0011, '\x79', False)
   conn.disconnect

def ledsgreen():
   conn.writeCharacteristic(0x0011, '\x7a', False)
   conn.disconnect

#-----------------------------------------------

# Slider Function
#-----------------------------------------------


def getsliderval():
   if (speed.get()==1):
    speed_1()
   elif (speed.get()==2):
    speed_2()
   elif (speed.get()==3):
    speed_3()
   elif (speed.get()==4):
    speed_4()
   elif (speed.get()==5):
    speed_5()
   elif (speed.get()==6):
    speed_6()
   elif (speed.get()==7):
    speed_7()
   elif (speed.get()==8):
    speed_8()
   elif (speed.get()==9):
    speed_9()

#-----------------------------------------------
-
# Buttons
#-----------------------------------------------
-
# Direction Buttons
#-----------------------------------------------
-
```

```
forwardbutton = Button(frame, text="Forward", comma
nd=forward, image=photo3)
forwardbutton.grid(row=1, column=1, padx=0, pady=5)


backbutton = Button(frame, text="Reverse",command=b
ack, image=photo4)
backbutton.grid(row=3, column=1, padx=0, pady=5)

rightbutton = Button(frame, text="Right", command =
right, image=photo2)
rightbutton.grid(row=2, column=2, padx=0, pady=0)

leftbutton = Button(frame, text="Left", command=lef
t,image=photo1)
leftbutton.grid(row=2, column=0, padx=0, pady=0)

stopbutton = Button(frame, text="Stop",  command=st
op, image=photo5)
stopbutton.grid(row=2, column=1, padx=0, pady=0)

#-----------------------------------------------
--
# Arm Buttons
#-----------------------------------------------
--

HolderUpbutton = Button(framearm,text="Up",command=
Holder_Up,image=photo3)
HolderUpbutton.grid(row=1, column=1, padx=10, pady=
5)

HolderDownbutton = Button(framearm,text="Down",comm
and=Holder_Down, image=photo4)
HolderDownbutton.grid(row=3, column=1, padx=10, pad
y=5)

stopbutton2 = Button(framearm, text="Stop", command
=stop, image=photo5)
stopbutton2.grid(row=2, column=1, padx=10, pady=5)

#-----------------------------------------------
```

```
--
# Grip Buttons
#-----------------------------------------------
--

HandClosebutton = Button(framearm, text="Close",com
mand=Hand_close,image=photo4)
HandClosebutton.grid(row=3, column=0, padx=10, pady
=5)

HandOpenbutton = Button(framearm,text="Open",comman
d=Hand_open,image=photo3)
HandOpenbutton.grid(row=1, column=0, padx=10, pady=
5)

stopbutton3 = Button(framearm, text="Stop", command
=stop, image=photo5)
stopbutton3.grid(row=2, column=0, padx=10, pady=5)

#-----------------------------------------------
--
# Speed Set Button
#-----------------------------------------------
--

setspeed = Button(framespeed, text="ENTER", command
=getsliderval,bg='#0066cc',fg='white')
setspeed.grid(row=2, column=0, padx=10, pady=10)

#-----------------------------------------------
--
# Range Button
#-----------------------------------------------
--

rangeonbutton = Button(framerange, text="Get Range
",  command=RangeOn, repeatinterval=5, repeatdelay
=5,bg='#0066cc',fg='white')
rangeonbutton.grid(row=1, column=0, padx=10, pady=1
0)

rangeoffbutton = Button(framerange, text="RangeOff
```

```
",  command=RangeOff,bg='#0066cc',fg='white')
rangeoffbutton.grid(row=3, column=0, padx=10, pady=
10)

ledsonbutton = Button(frameleds, text="Leds On",
command=ledson, repeatinterval=5, repeatdelay=5,bg
='#0066cc',fg='white')
ledsonbutton.grid(row=1, column=0, padx=10, pady=10
)

ledsoffbutton = Button(frameleds, text="Leds Off",
  command=ledsoff,bg='#0066cc',fg='white')
ledsoffbutton.grid(row=3, column=0, padx=10, pady=1
0)

ledsbluebutton = Button(frameleds, text="Blue",  c
ommand=ledsblue,bg='#0066cc',fg='white')
ledsbluebutton.grid(row=4, column=0, padx=10, pady=
10)

ledsredbutton = Button(frameleds, text="Red",  comm
and=ledsred,bg='#0066cc',fg='white')
ledsredbutton.grid(row=5, column=0, padx=10, pady=1
0)

ledsgreenbutton = Button(frameleds, text="Green",
command=ledsgreen,bg='#0066cc',fg='white')
ledsgreenbutton.grid(row=6, column=0, padx=10, pady
=10)

servo0button = Button(frameservo, text="Front",  co
mmand=servo0,bg='#0066cc',fg='white')
servo0button.grid(row=1, column=0, padx=10, pady=10
)

servo90button = Button(frameservo, text="Right",
command=servo90,bg='#0066cc',fg='white')
servo90button.grid(row=2, column=0, padx=10, pady=1
0)

servo180button = Button(frameservo, text="Left",  c
ommand=servo180,bg='#0066cc',fg='white')
```

```
servo180button.grid(row=3, column=0, padx=10, pady=
10)

root.mainloop()
```