pyduino.py

```
"""
A library to interface Arduino through serial connection
"""
import serial


class Arduino():
 """
 Models an Arduino connection
 """


 def __init__(self, serial_port='/dev/ttyACM0', baud_rate=9600,
     read_timeout=5):
   """
   Initializes the serial connection to the Arduino board
   """
   self.conn = serial.Serial(serial_port, baud_rate)
   self.conn.timeout = read_timeout # Timeout for readline()


 def set_pin_mode(self, pin_number, mode):
   """
   Performs a pinMode() operation on pin_number
   Internally sends b'M{mode}{pin_number} where mode could be:
   - I for INPUT
   - O for OUTPUT
   - P for INPUT_PULLUP MO13
   """
   command = (''.join(('M',mode,str(pin_number)))).encode()
   #print 'set_pin_mode =',command,(''.join(('M',mode,str(pin_number))))
```

```
    self.conn.write(command)

def digital_read(self, pin_number):
    """
    Performs a digital read on pin_number and returns the value (1 or 0)
    Internally sends b'RD{pin_number}' over the serial connection
    """
    command = (''.join(('RD', str(pin_number)))).encode()
    self.conn.write(command)
    line_received = self.conn.readline().decode().strip()
    header, value = line_received.split(':') # e.g. D13:1
    if header == ('D'+ str(pin_number)):
        # If header matches
        return int(value)

def digital_write(self, pin_number, digital_value):
    """
    Writes the digital_value on pin_number
    Internally sends b'WD{pin_number}:{digital_value}' over the serial
    connection
    """
    command = (''.join(('WD', str(pin_number), ':',
        str(digital_value)))).encode()
    self.conn.write(command)

def analog_read(self, pin_number):
    """
    Performs an analog read on pin_number and returns the value (0 to 1023)
    Internally sends b'RA{pin_number}' over the serial connection
    """
    command = (''.join(('RA', str(pin_number)))).encode()
    self.conn.write(command)
    line_received = self.conn.readline().decode().strip()
    header, value = line_received.split(':') # e.g. A4:1
    if header == ('A'+ str(pin_number)):
        # If header matches
        return int(value)

def analog_write(self, pin_number, analog_value):
    """
    Writes the analog value (0 to 255) on pin_number
    Internally sends b'WA{pin_number}:{analog_value}' over the serial
    connection
    """
    command = (''.join(('WA', str(pin_number), ':',
        str(analog_value)))).encode()
    self.conn.write(command)


def close(self):
    """
    To ensure we are properly closing our connection to the
    Arduino device.
    """
    self.conn.close()
    print 'Connection to Arduino closed'
```