import i2c\_lib

from time import sleep

# LCD Address

#ADDRESS = 0x3F

ADDRESS = 0x27

# I2C bus

BUS = 1

# commands

LCD\_CLEARDISPLAY = 0x01

LCD\_RETURNHOME = 0x02

LCD\_ENTRYMODESET = 0x04

LCD\_DISPLAYCONTROL = 0x08

LCD\_CURSORSHIFT = 0x10

LCD\_FUNCTIONSET = 0x20

LCD\_SETCGRAMADDR = 0x40

LCD\_SETDDRAMADDR = 0x80

# flags for display entry mode

LCD\_ENTRYRIGHT = 0x00

LCD\_ENTRYLEFT = 0x02

LCD\_ENTRYSHIFTINCREMENT = 0x01

LCD\_ENTRYSHIFTDECREMENT = 0x00

# flags for display on/off control

LCD\_DISPLAYON = 0x04

LCD\_DISPLAYOFF = 0x00

LCD\_CURSORON = 0x02

LCD\_CURSOROFF = 0x00

LCD\_BLINKON = 0x01

LCD\_BLINKOFF = 0x00

# flags for display/cursor shift

LCD\_DISPLAYMOVE = 0x08

LCD\_CURSORMOVE = 0x00

LCD\_MOVERIGHT = 0x04

LCD\_MOVELEFT = 0x00

# flags for function set

LCD\_8BITMODE = 0x10

LCD\_4BITMODE = 0x00

LCD\_2LINE = 0x08

LCD\_1LINE = 0x00

LCD\_5x10DOTS = 0x04

LCD\_5x8DOTS = 0x00

# flags for backlight control

LCD\_BACKLIGHT = 0x08

LCD\_NOBACKLIGHT = 0x00

En = 0b00000100 # Enable bit

Rw = 0b00000010 # Read/Write bit

Rs = 0b00000001 # Register select bit

class lcd:

"""

Class to control the 16x2 I2C LCD display from sainsmart from the Raspberry Pi

"""

def \_\_init\_\_(self):

"""Setup the display, turn on backlight and text display + ...?"""

self.device = i2c\_lib.i2c\_device(ADDRESS, BUS)

self.write(0x03)

self.write(0x03)

self.write(0x03)

self.write(0x02)

self.write(LCD\_FUNCTIONSET | LCD\_2LINE | LCD\_5x8DOTS | LCD\_4BITMODE)

self.write(LCD\_DISPLAYCONTROL | LCD\_DISPLAYON)

self.write(LCD\_CLEARDISPLAY)

self.write(LCD\_ENTRYMODESET | LCD\_ENTRYLEFT)

sleep(0.2)

def strobe(self, data):

"""clocks EN to latch command"""

self.device.write\_cmd(data | En | LCD\_BACKLIGHT)

sleep(0.0005)

self.device.write\_cmd(((data & ~En) | LCD\_BACKLIGHT))

sleep(0.001)

def write\_four\_bits(self, data):

self.device.write\_cmd(data | LCD\_BACKLIGHT)

self.strobe(data)

def write(self, cmd, mode=0):

"""write a command to lcd"""

self.write\_four\_bits(mode | (cmd & 0xF0))

self.write\_four\_bits(mode | ((cmd << 4) & 0xF0))

def display\_string(self, string, line):

if line == 1:

self.write(0x80)

if line == 2:

self.write(0xC0)

if line == 3:

self.write(0x94)

if line == 4:

self.write(0xD4)

for char in string:

self.write(ord(char), Rs)

def clear(self):

"""clear lcd and set to home"""

self.write(LCD\_CLEARDISPLAY)

self.write(LCD\_RETURNHOME)

def backlight\_off(self):

"""turn off backlight, anything that calls write turns it on again"""

self.device.write\_cmd(LCD\_NOBACKLIGHT)

def display\_off(self):

"""turn off the text display"""

self.write(LCD\_DISPLAYCONTROL | LCD\_DISPLAYOFF)

def display\_on(self):

"""turn on the text display"""

self.write(LCD\_DISPLAYCONTROL | LCD\_DISPLAYON)