

Electronics Serial Communication-SPI

Terry Sturtevant

Wilfrid Laurier University

November 16, 2016

Serial Communication -SPI

Serial Communication -SPI

- Serial Peripheral Interface

Serial Communication -SPI

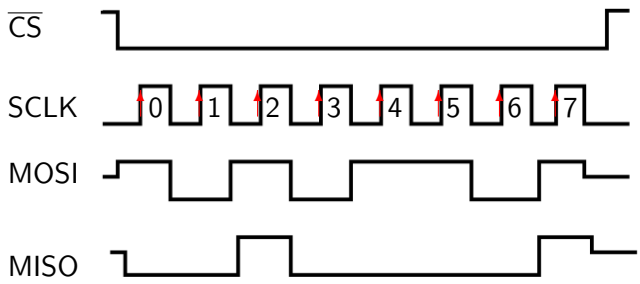
- Serial Peripheral Interface
- Master/slave communication

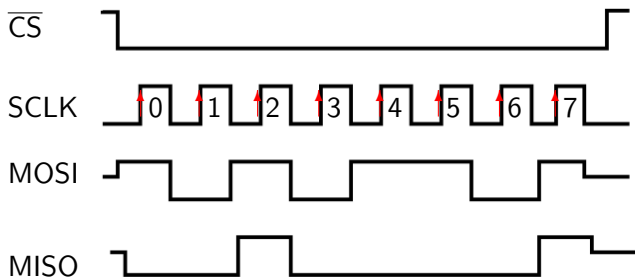
Serial Communication -SPI

- Serial Peripheral Interface
- Master/slave communication
- Uses 3 signals (and Ground),
MISO, MOSI, SCLK
and chip selects for each slave device

Serial Communication -SPI

- Serial Peripheral Interface
- Master/slave communication
- Uses 3 signals (and Ground),
MISO, MOSI, SCLK
and chip selects for each slave device
- Synchronous, so master controls clock rate





SPI transfers can happen in both directions simultaneously.

PySpidev

PySpidev

- **spi = spidev.SpiDev()**
create object

PySpidev

- **spi = spidev.SpiDev()**
create object
- **spi.open(0,0)**
open port, device

PySpidev

- **spi = spidev.SpiDev()**
create object
- **spi.open(0,0)**
open port, device
- **response = spi.xfer2([0xAA])**
transfer one byte
CS held active between blocks

PySpidev

- **spi = spidev.SpiDev()**
create object
- **spi.open(0,0)**
open port, device
- **response = spi.xfer2([0xAA])**
transfer one byte
CS held active between blocks
- **response = spi.xfer([values])**
transfer bytes
CS released and reactivated between blocks

PySpidev (continued)

PySpidev (continued)

- **spi.writebytes([values])**
write bytes

PySpidev (continued)

- **spi.writebytes([values])**
write bytes
- **spi.readbytes(len)**
read *len* bytes

PySpidev (continued)

- **spi.writebytes([values])**
write bytes
- **spi.readbytes(len)**
read *len* bytes
- **spi.cshigh**
get or set active state of CS

PySpidev (continued)

- **spi.writebytes([values])**
write bytes
- **spi.readbytes(len)**
read *len* bytes
- **spi.cshigh**
get or set active state of CS
- **spi.close()**
close port

PySpidev sample code

PySpidev sample code

```
import spidev
spi = spidev.SpiDev()
spi.open(0,0) #port , device
# use chip select CE0, channel 0
while True:
    strval = raw_input(" val (0...255 , q=quit):")
    if strval == 'q':
        break
    else:
        value = int (strval)
        dummy = spi.xfer2 ([49 , value])
spi.close()
```