Introduction to Robotics

Lecture 6: Sensors and Communication Buses

22. 10. 2018

ParaDiSe

Today Goal

Theory:

- get overview about sensors
- get overview about communication buses

Practise:

- use the line sensor

Sensors

What Can We Measure In The Digital World?

Using purely digital devices:

- digital 1 or 0
- time (using clock source and timers)
- frequency of a digital signal

This is not enough.

Analog To Digital Converter (ADC)

- peripheral/device measuring voltage in given range
- usually a part of a microcontroler (Arduino: `analogRead`)

Parameters:

- resolution (8–24bit)
- speed (100 S/s up to 10 GS/s)
- accuracy

Measuring Electric Quantities

- current – voltage drop on a resistor
- capacity – frequency in oscillator
- capacity – impedance in AC
- resistance – voltage divider, drop using known current
- power – voltage $\times$ current

Measuring Proximity and Distance

- mechanical switch
- IR-reflection (phototransistor $\rightarrow$ voltage)
- ultrasonic sensor
- triangulative LIDAR
- TOF LIDAR

Other Physical Quantities

- termistor – resistance change in conductor
- temperature – thermocouple (two bond metals generate voltage)
- magnetic field – Hall sensor (voltage)
- light – phototransistor, photodiode, photo resistor
- pressure
- humidity
- tenzometr interrupt

Accelerometer vs. Gyroscope

Accelerometer

- measures acceleration (X, Y, Z)
- small weights on spring attached to a capacitor

Gyroscope

- measures angular velocity (yaw, pitch, roll)
- Corriolis force

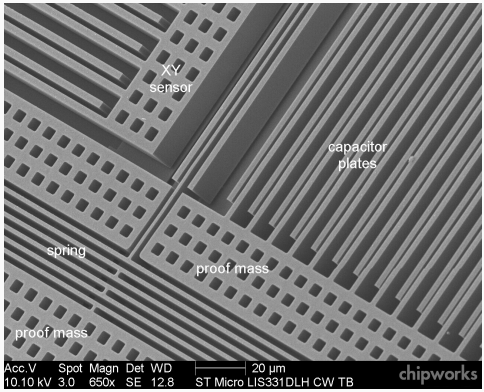


Figure 1: source:
http://www.memsjournal.com/2010/12/motion-sensing-in-the-iphone-4-mems-accelerometer.html

Modern Sensor

Sensors are hard to master:

- complex analog circuity
- noisy
- calibration & accuracy

Modern sensors:

- sensor + analog circuity + digital circuity in a single components
- calibration done by vendor
- provide digital values
- communication using a bus

Communication Buses

Communication Buses in General

- usually as simple as possible
- timing might be crucial
- all common buses have a peripheral in MCU
- bit-bang = to implement a bus in software (slow)

Serial Line

- no clock $\rightarrow$ communication speed defined in advance
- no master/slave
- only two device can communicate
- serial line != RS232

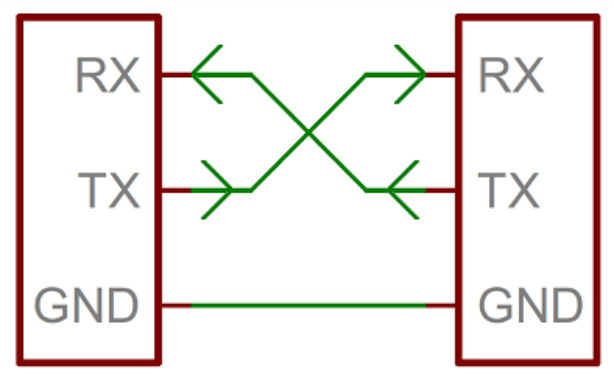


Figure 2: source: `https://learn.sparkfun.com/tutorials/serial-communication`

Serial Line

Figure 3: source: `https://learn.sparkfun.com/tutorials/serial-communication`

Serial Peripheral Interface (SPI)

- synchronous, duplex
- slave select
- can be high speed (usually 5Mhz, up to 100 MHz)
- multiple slaves, single master

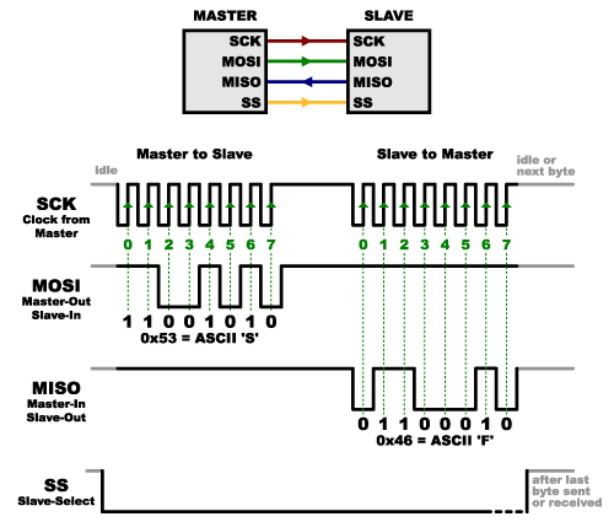


Figure 4: source: https://learn.sparkfun.com/

Serial Peripheral Interface (SPI)

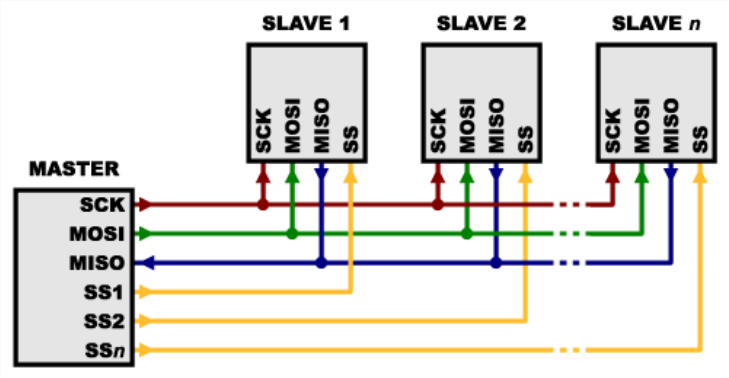


Figure 5: source:
`https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi`

SPI Example

```
#include <Arduino.h>
#include <SPI.h>
void setup() {
    pinMode(ss, OUTPUT);
    SPI.begin();
}
uint8_t setAndReadValue(int value) {
    uint8_t ret;
    digitalWrite(ss, LOW);
    SPI.transfer( value ); ret = SPI.transfer( 0 );
    digitalWrite(ss, HIGH);
    return ret;
}
```

Inter-Integrated Circuit (I2C)

- synchronous, half-duplex
- each device has address
- slow (10 kHz-400kHz, special HW 1 Mhz)
- multiple slaves, multiple masters

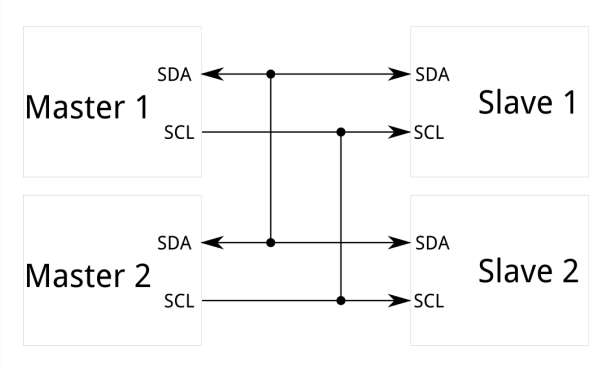


Figure 6: source: `https://learn.sparkfun.com/tutorials/i2c`

Inter-Integrated Circuit (I2C)

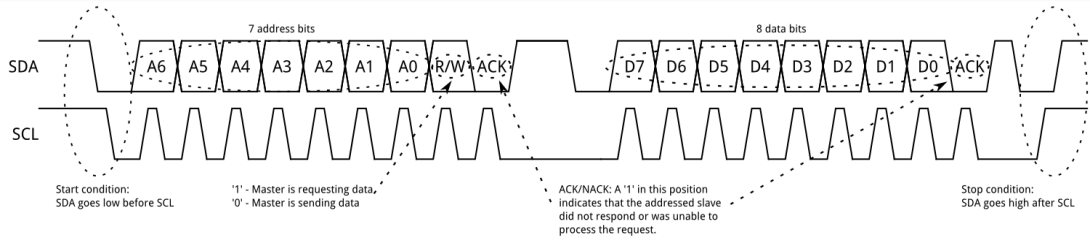
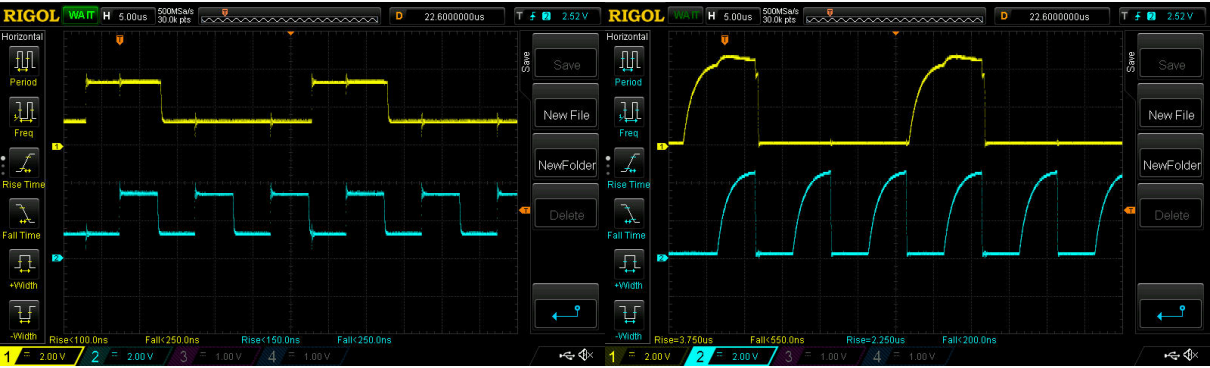


Figure 7: source: `https://learn.sparkfun.com/tutorials/i2c`

I2C Example

```
#include <Arduino.h>
#include <Wire.h>
void setup() {
    Wire.begin();
    Serial.begin(9600);
}
void loop() {
    Wire.requestFrom(8, 6);    // request 6 bytes from slave device #8
    while (Wire.available()) { // slave may send less than requested
        char c = Wire.read();
        Serial.print(c);
    }
    delay(500);
}
```

Why Is I2C Slow?



Source `https://www.sparkfun.com/news/2366`

Practical Part

Our Line Sensor

- 8 IR sensors
- SPI ADC - MCP3008
- values in range 0-1024

- use the Arduino SPI library & read the datasheet or
- use the Adafruit MCP3008 library (add `lib_deps = Adafruit MCP3008` in your platformio.ini)
- write a program which outputs 8 comma separated values on serial line at 9600 bauds
- call `line_preview.py /dev/ttyUSB0` and play