CSE121: IoT

debug + i2C

Jose Renau
renau@ucsc.edu

UCSC Baskin School of Engineering
University of California, Santa Cruz

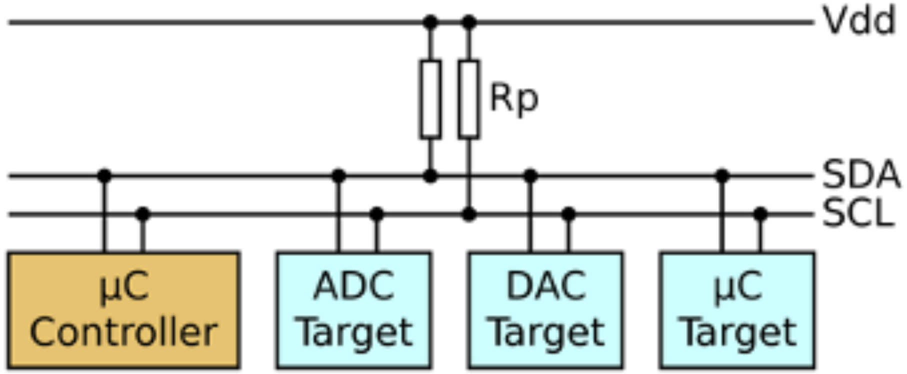
Announcements

- Lab1 due Tomorrow
- Lab2 released Today after class
- Next class we have our first quiz

- CSE121 times:
  - Monday: John noon-2pm
  - Tuesday: class 9:50-11:25, John noon-2pm
  - Wednesday: Aravind 3-5pm
  - Thursday: class 9:50-11:25, Jose 3-4:30pm
  - Friday: Aravind: 10am-noon

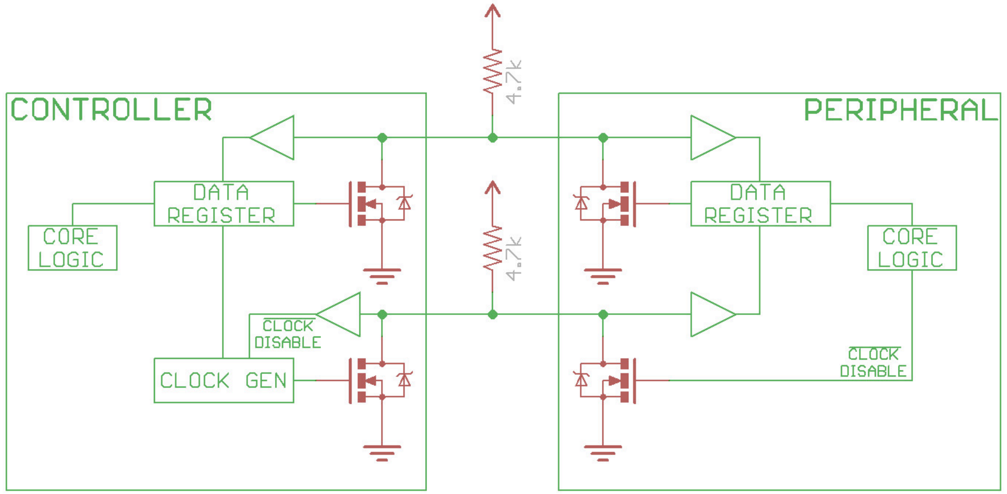
I2C

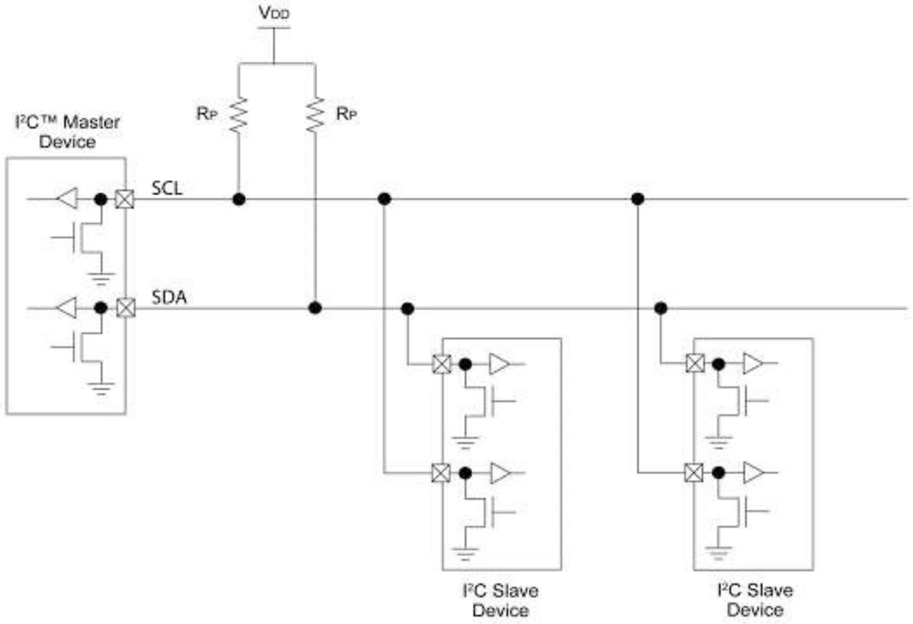
- I2C: Inter Integrated Circuit
- Invented by Philips in 1982
- 2 bidirectional open-drain lines
  - SCL: clock
  - SDA: data
- Different Speeds
  - 400kbit (common)
  - 5Mbit (ultra-fast)
- To further reduce wires, I2C operates at a bus
  - One Controller per bus



I2C: Electrical considerations

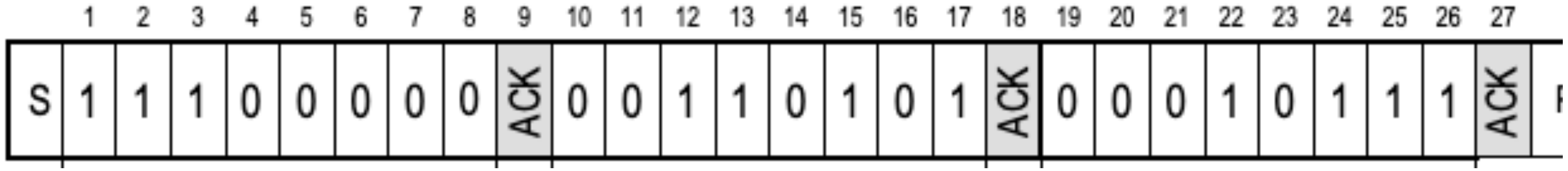
- I2C tends to be 3.3 or 5V (chip dependent)
- 100KHz clock (typical), ESP32 seems to support 400KHz
- Many devices can connect on the same line
- Using ESP32, you do not need to add the resistors to the breadboard





I2C

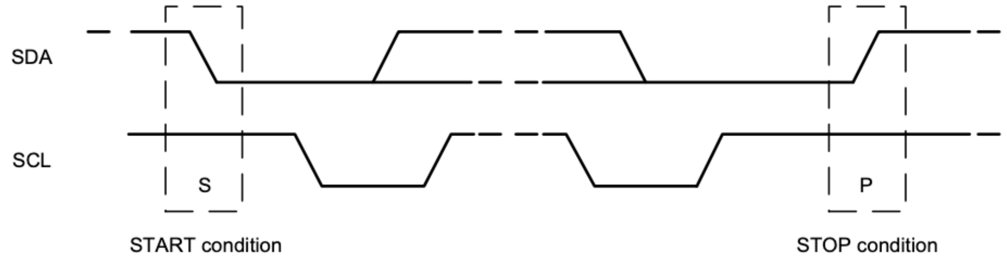
- 2 Modes
  - Controller
  - Peripheral
- 7-bit device address
  - Up to 128 devices on a bus
- Data is transmitted in the form of packets
  - Start - 1 bit
  - Data- 8 bit (first bit is R/W bit)
  - Start - 1 bit
  - Acknowledge - 1 bit (from the peripheral)

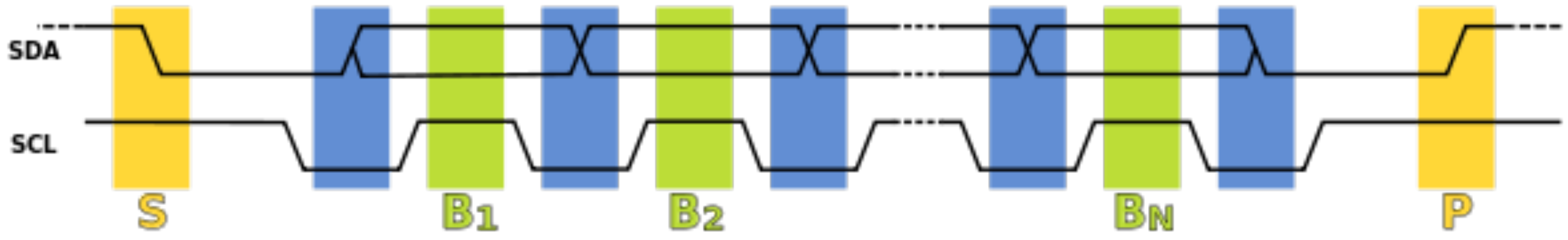


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
S	1	1	1	0	0	0	0	0	ACK	0	0	1	1	0	1	0	1	ACK	0	0	0	1	0	1	1	1	ACK	

I2C: Timing Diagram

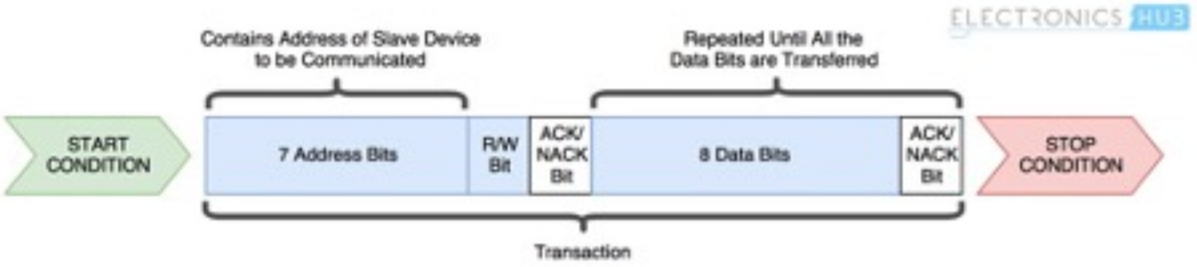
- Start: SDA low while SCL stays high
- Data transmission phase. Data change while SCL is low, sampled when SCL is high
- Peripheral needs to ACK every byte (SDA low)
- Stop: SDA high while SCL stays high
  - I2C defines how to transfer data. The meaning of data is device specific!





I2C

- 8-bit data packet
  - First bit is always Read/write
  - The first 7 data bits address the device
  - The next packet’s data is device specific



- Data packet Example
  - First byte specifies register address (128 registers)
  - Second byte specifies data written to register
- Fully Device specific (needs a device driver)
  - E.g. can use many address bytes for enabling larger memory/register spaces

I2C: Clock Stretching

- Clock stretching
  - Peripheral device can inform controller that it cannot absorb additional data
  - Pull down SCL, which can be sensed by controller
  - Makes SCL bidirectional
- Buffering and Multiplexing
  - 128 devices on the same bus introduces high capacitance
  - Prohibits high clock speed
  - Buffer chips to refresh signal

I3C

- New 2-wire interface (still SDA/SCLK) but
  - Higher at a rate (20MHz)
  - Dynamic address allocation (not fix per chip)
  - CRC
  - Interrupts
  - …
  -

Let’s debug something

- `idf.py create-project test1`
- `idf.py openocd`

- Run gdb

```
riscv32-esp-elf-gdb -x gdbinit build/xxx.elf
```

- Gdbinit:

```
target remote :3333
set remote hardware-watchpoint-limit 2
mon reset halt
flushregs
thb app_main
c
```

Next Class

- PWM