UAV Workshop

Quadrotor Components

Prof. Venki Muthukumar, Ph.D.

UAS: Unmanned Aircraft (Quadrotor)

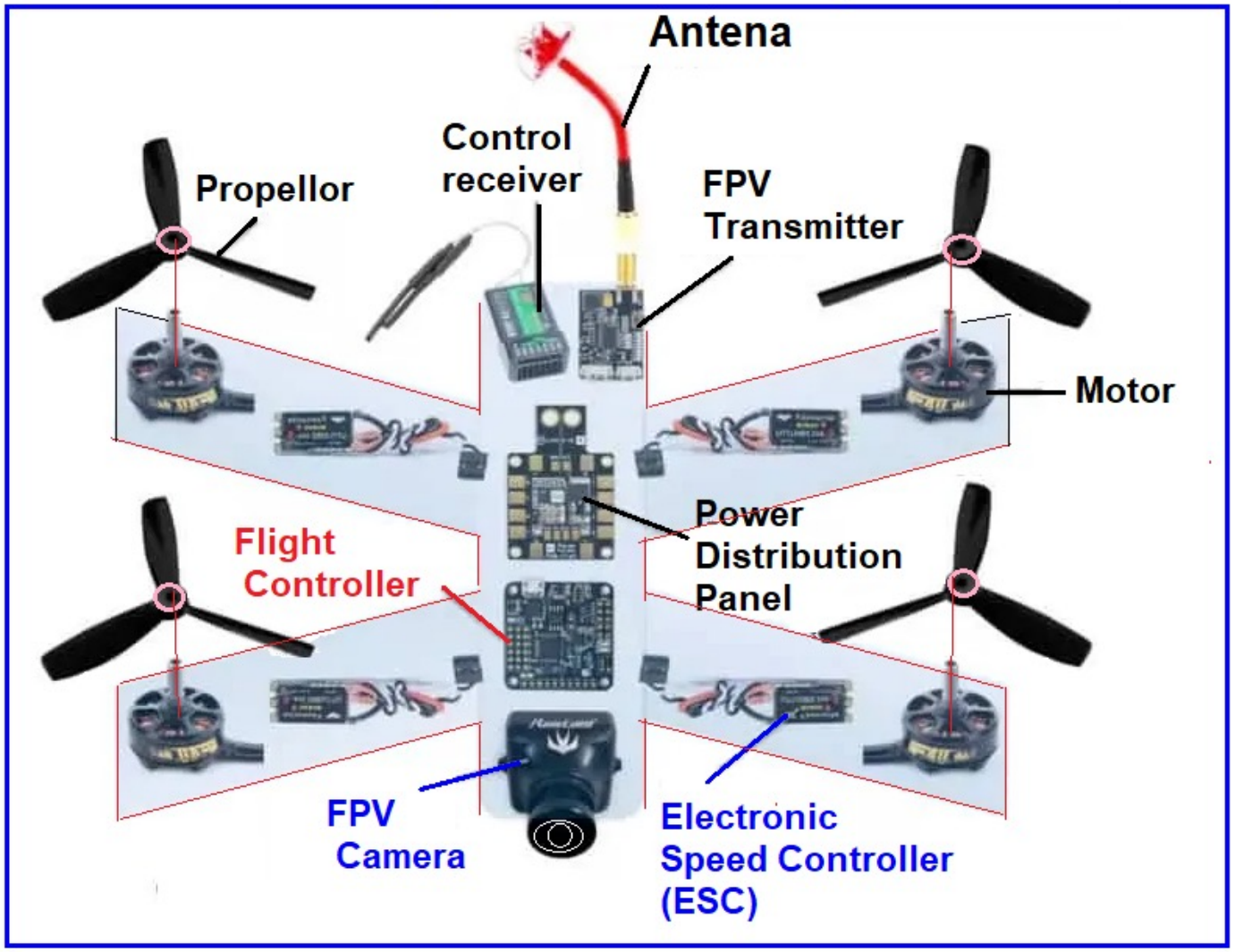
- **UA Hardware**
  - Motors
  - Propellers
  - Electronics
    - Avionics System
      - Control CPU (Autopilot Units)
      - Inertial Measurement Unit (IMU)
      - Electronic Speed Controller (ESC)
      - Radio-Frequency (RF) Receivers
    - Power Supply
      - Lithium-Polymer Battery (Li-Po) - in most Micro/Mini UAV's
    - Sensors (for mission)
      - Cameras (electro-optical or infrared)
      - Lasers (targeting or depth measure)

Table 11.5 Weight budget for quadrotor MAV

Components	Amount	Estimate weight (g)	Current weight (g)
Battery	1	10	9.8
Motor and propeller	4	3.5	3.59
Quadrotor arm	4	1	0.93
Quadrotor frame	1	4	2.13
Onboard system	1	10	7.32
Miscellaneous		2	1.7
Total		44	39.03

http://ardupilot.org/copter/docs/hoverthings-flip-sport-quadcopter.html

Components of a Quadrotor

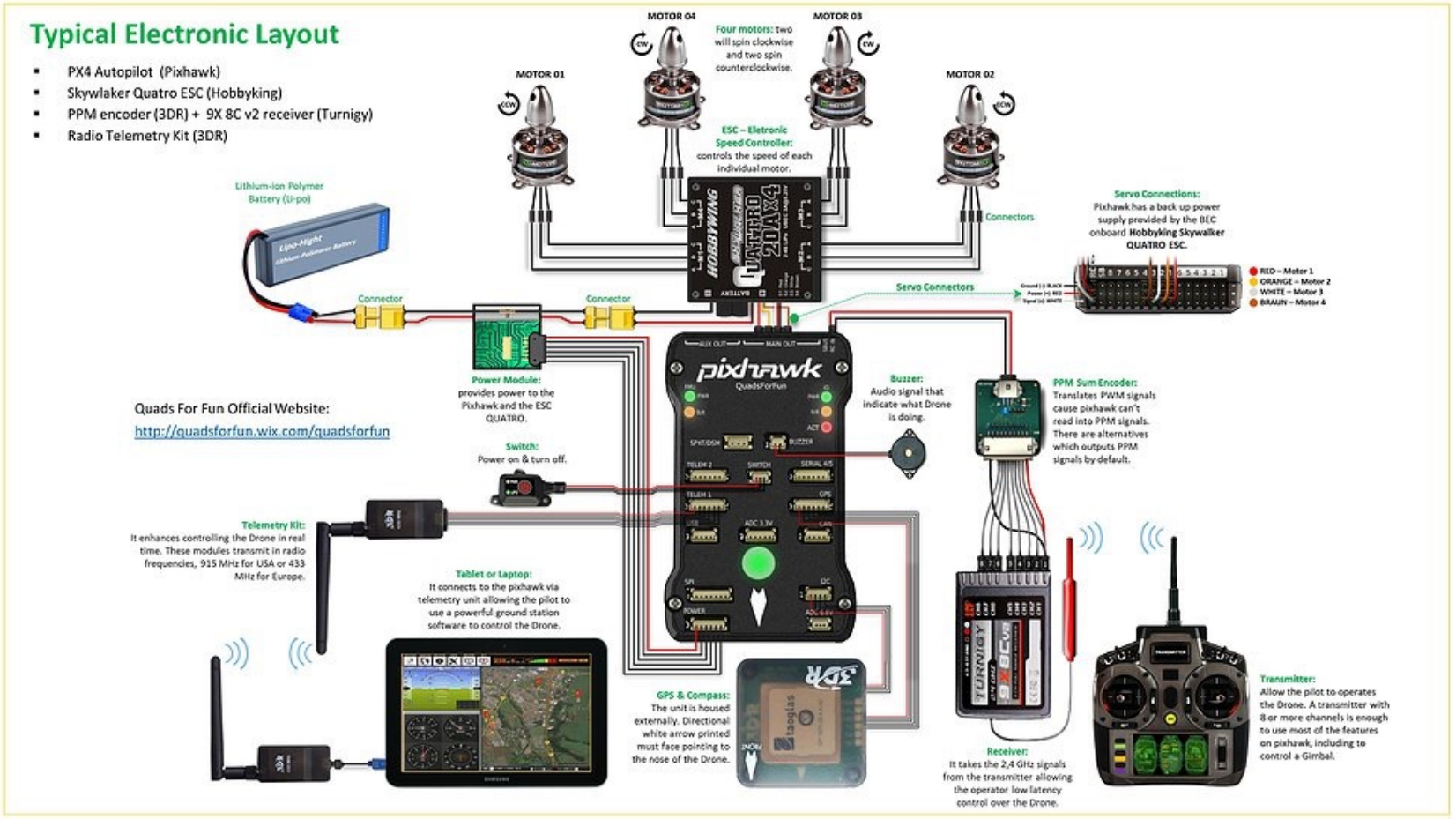


UAS: Unmanned Aircraft (Micro/Mini) Vehicle

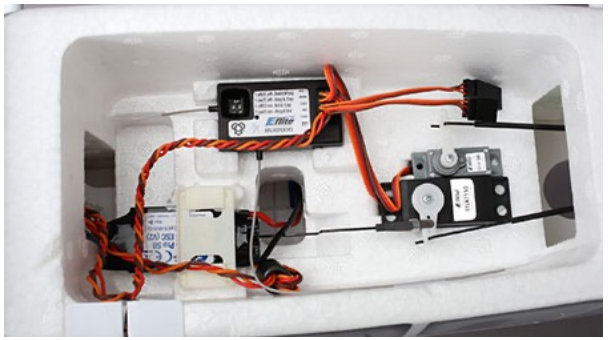
UAV Workshop

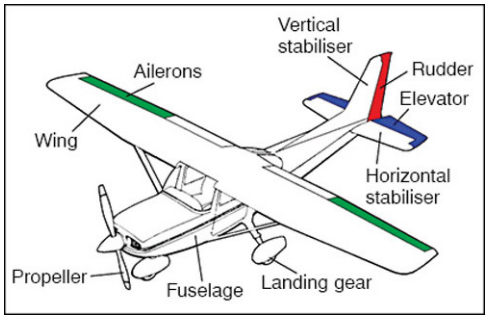
Typical Electronic Layout

- PX4 Autopilot (Pixhawk)
- Skywlaker Quatro ESC (Hobbyking)
- PPM encoder (3DR) + 9X 8C v2 receiver (Turnigy)
- Radio Telemetry Kit (3DR)

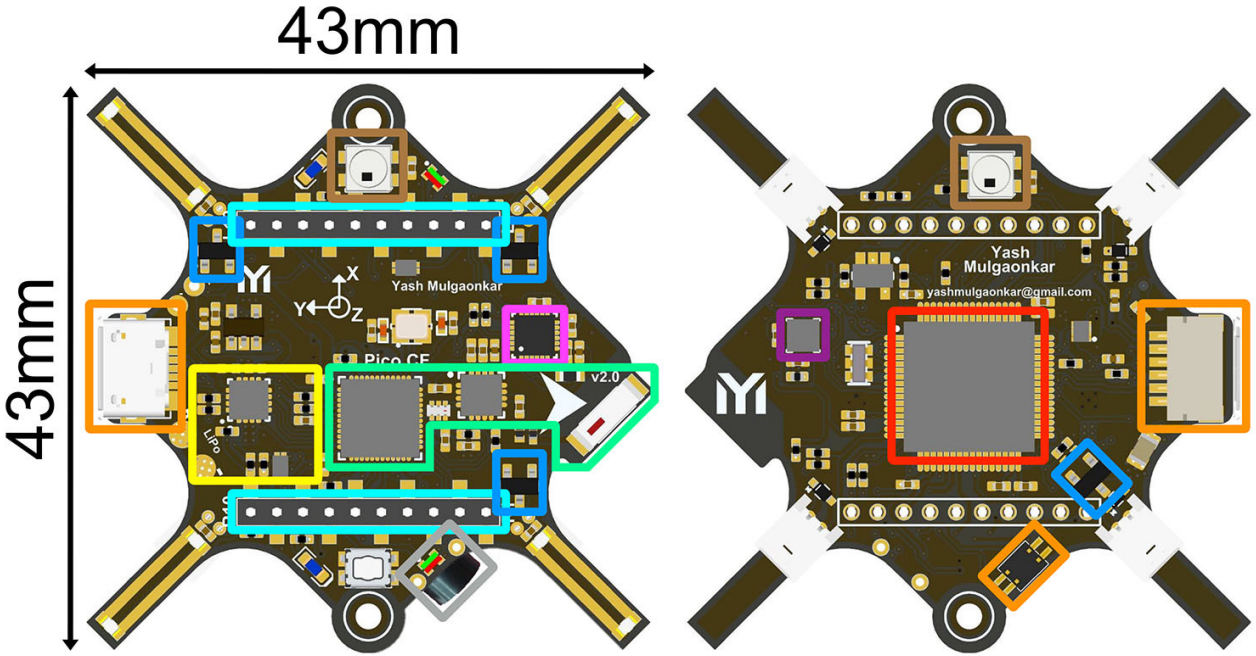


UAS: Unmanned Aircraft (FWA)



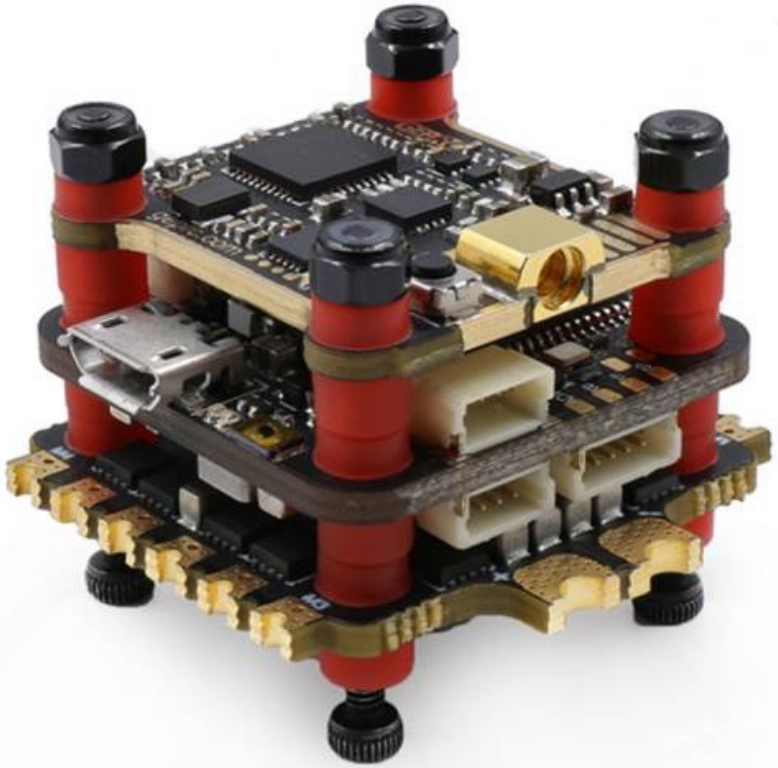


Autopilot



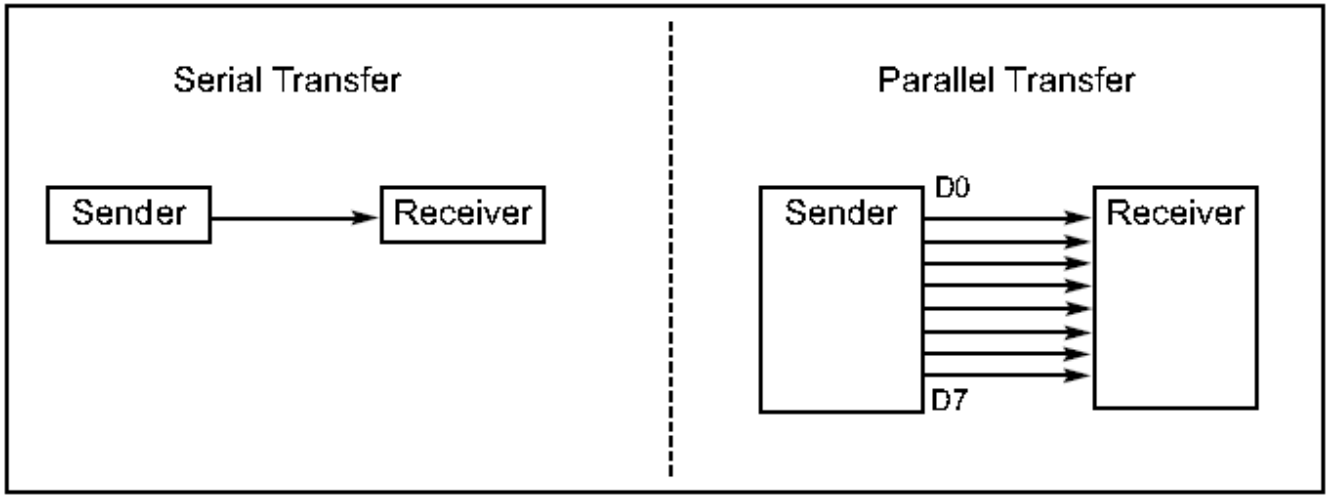
- ARM Cortex M4 Processor
- 2.4GHz Transceiver + Front-End
- MPU-9250 IMU
- Power Management Circuit
- High Resolution Barometer
- 4x Motor Drivers
- RGB LEDs
- Expansion Ports
- USB + SWD Ports
- Piezoelectric Buzzer

All in one Modules



Parallel vs. Serial

- Parallel
  - Transfer a byte of data at a time -> faster, easier
- Serial
  - Transfers a bit after another -> cheaper, ideal for long distance through phone line (modem is needed)



Direction

- Simplex: data can moves only in one directions
- Half Duplex: data can moves in two direction but not at the same time
- Full Duplex: data can moves in two direction at the same time

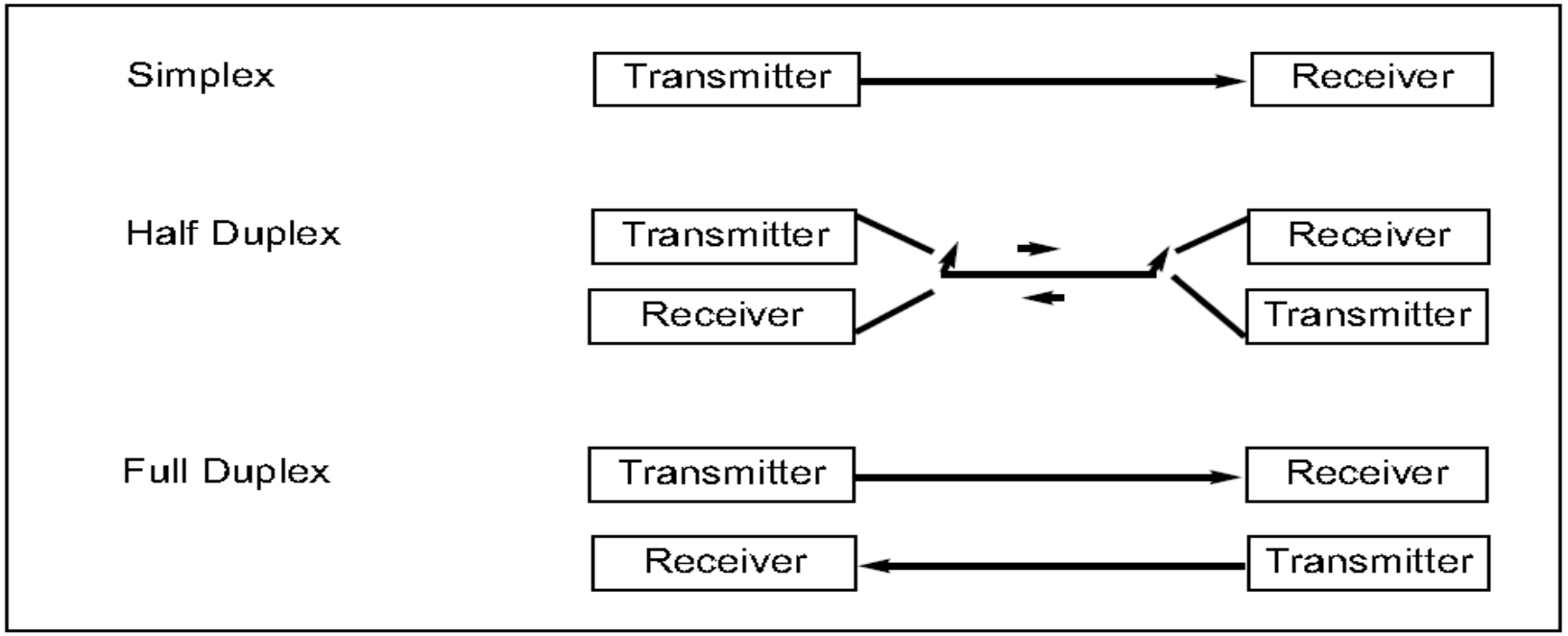
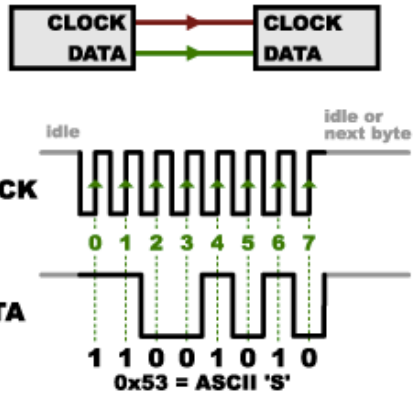


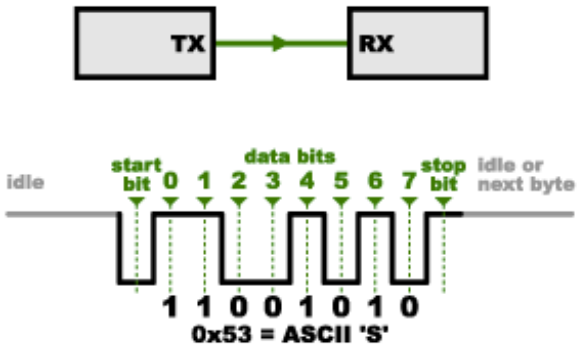
Figure 11-2. Simplex, Half-, and Full-Duplex Transfers

Synchronous vs. asynchronous

- Synchronous
  - Clock pulse should be transmitted during data transmission.
  - Only one side generates clock at the same time.



- Asynchronous
  - Clock pulse is not transmitted.
  - The two sides should generate clock pulse.
  - There should be a way to synchronize the two sides.



Serial Interface/UART

- Commonly used for one-to-one communication.
- There are many variants, the simplest uses just two lines, TX (transmit) and RX (receive).
- Transmission process (9600 baud, 1 bit=1/9600=0.104 mS), Numbers of bit transmitted in a second is called bps
- To synchronize the two sides, framing is used:
  - Each frame starts with a space (0) which is called Start bit
  - A character of 7-9 bits is transmitted after start bit
  - [a bit of parity can be transmitted after the character] (optional)
  - Each frame is ended by one or two mark (1) which is called stop bit(s).

SERIAL – ASYNCHRONOUS – TRANSMISSION

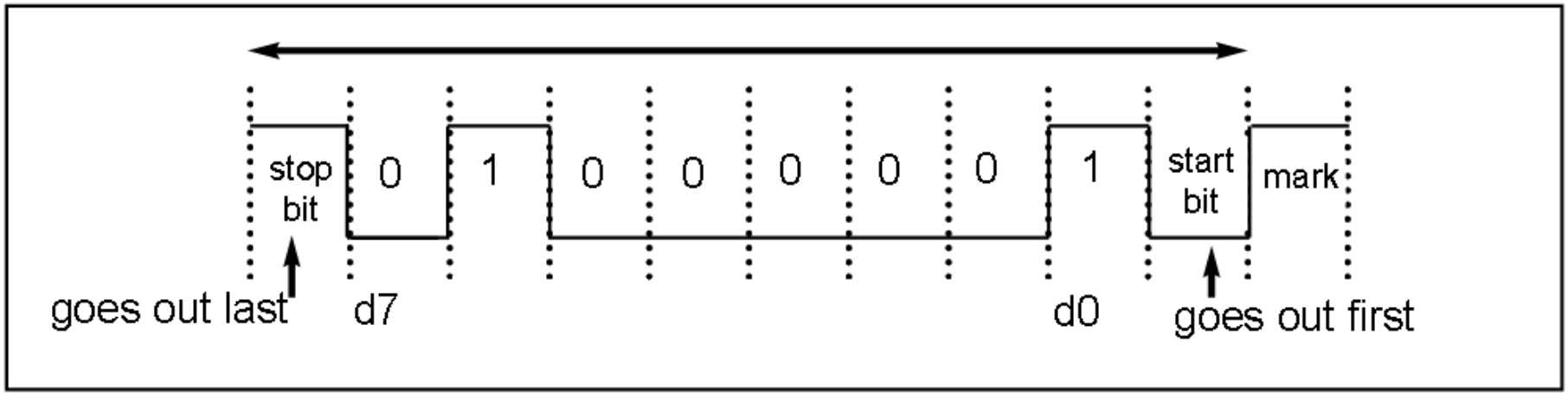
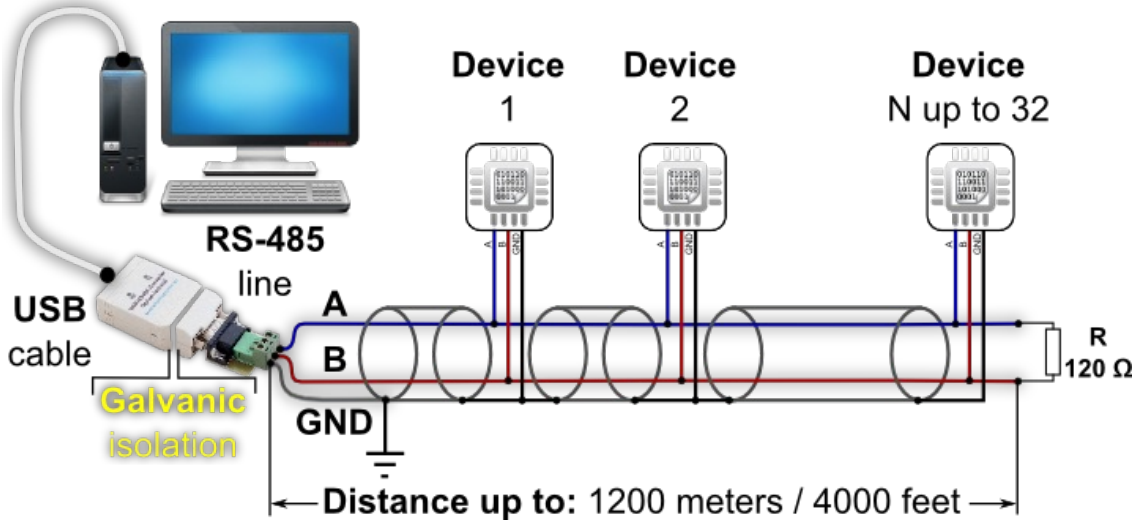


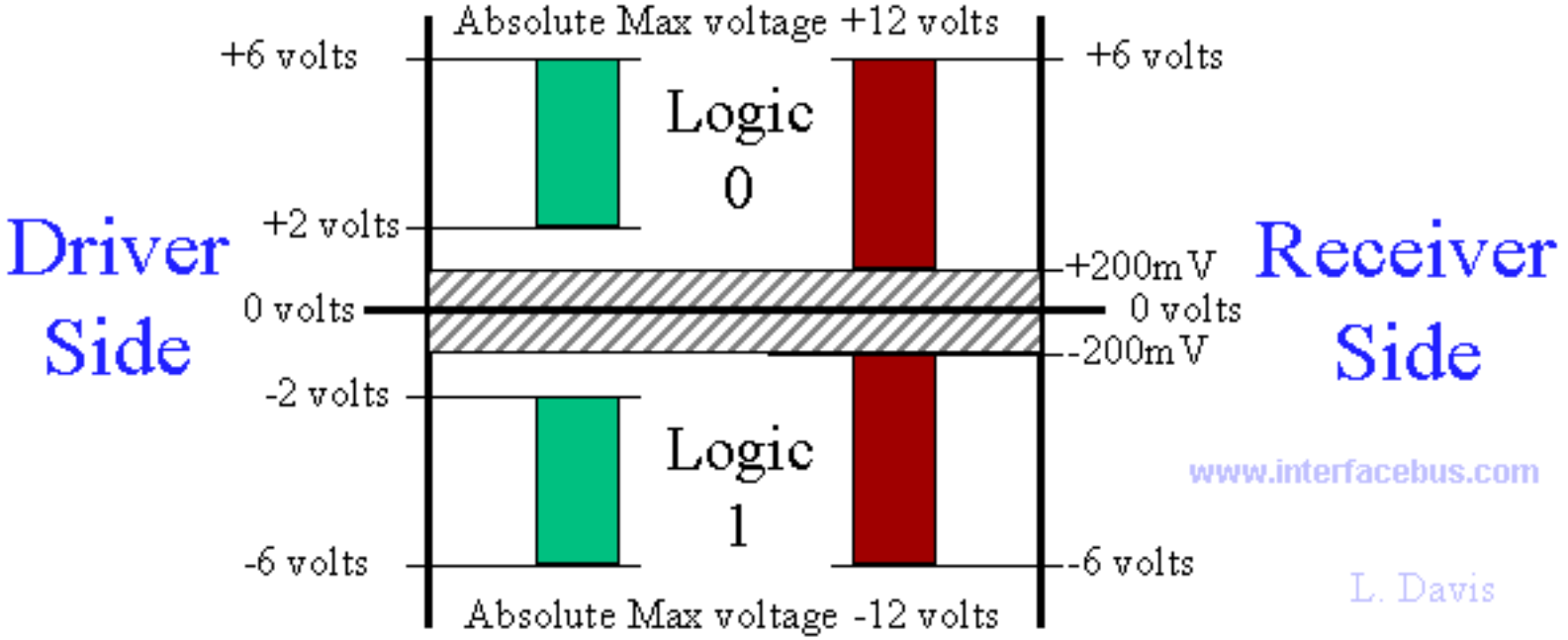
Figure 11-3. Framing ASCII 'A' (41H)

RS-485

- long cabling distances in electrically noisy environments and can support multiple devices on the same bus.
- tri-stable drivers that are separate programmable.
- only one device will work at a given time.



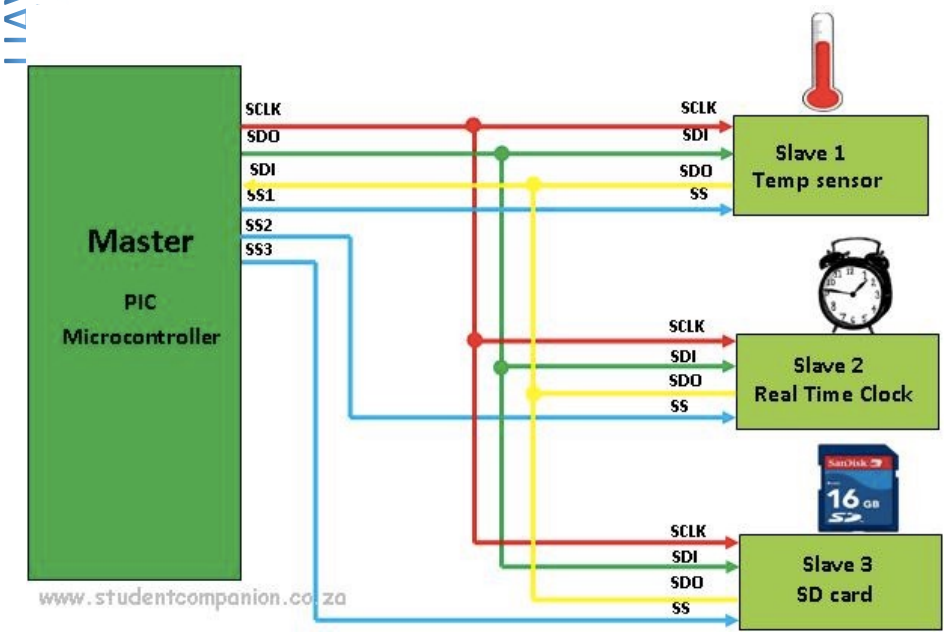
RS-485

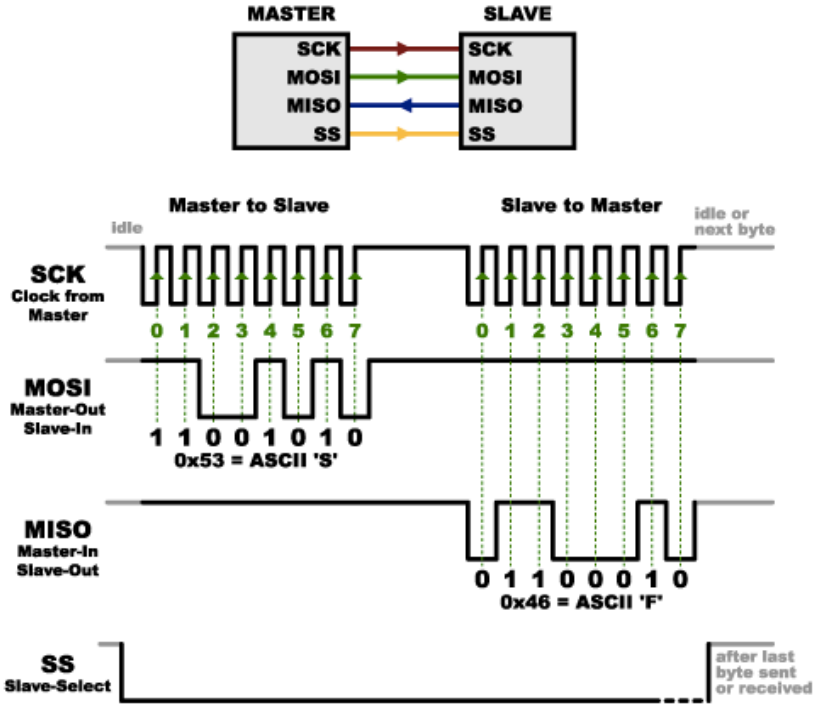


http://www.ti.com/lit/an/slla070d/slla070d.pdf

SPI Basics

- A communication protocol using 4 wires
  - Also known as a 4-wire bus
- Used to communicate across small distances
- Multiple Slaves, Single Master
- Synchronized





Inter Integrated Chip (IIC – I2C)

- The name I2C is shorthand for Standard Inter-Integrated Circuit bus
- I2C is a serial data protocol which operates with a master/slave relationship
- I2C only uses two physical wires, this means that data only travels in one direction at a time.
- I2C - Slaves Addressable
- Four max speeds(100kbS(standard),400kbS(fast),1 MbS (fast plus), and 3.4 MbS (high-speed)

I2C Write a Single Byte

1. All: allow SDA, SCL start high
2. Master: SDA low to signal start
3. Master: Send out SCL, and 7 bit address followed by 0 (~W) on SDA
4. Slave: Pull SDA low to signify ACKnowledge
5. Master: Send out 8 data bits on SDA
6. Slave: Ack
7. All: allow SDA to go high when SCL is high (stop)

- **For "Read",**
  3. Master: Address following by 1 (R) on SDA
  5. Slave: Send out 8 data bits on SDA
  6. Master: Ack

I2C