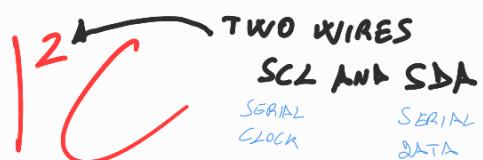
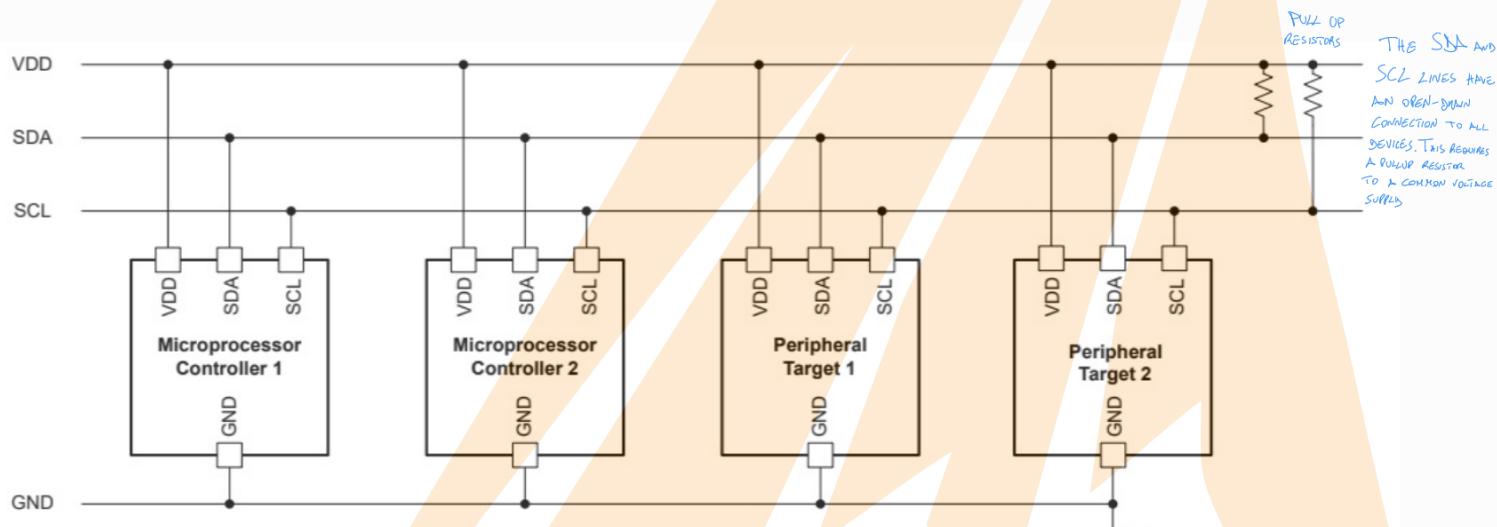


COMMUNICATION IS SENT IN BYTE PACKET FORM & UNIQUE ADDRESS FOR EACH TARGET DEVICE

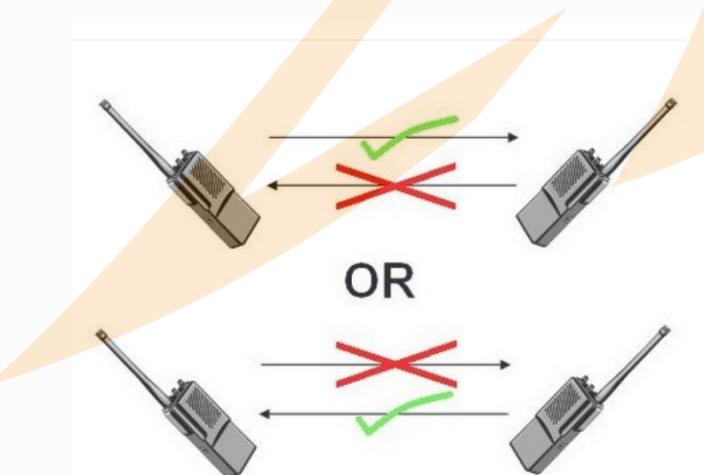


BOTH MASTER AND SLAVE CAN SEND AND RECEIVE DATA

I<sup>2</sup>C IS A TWO-WIRE SERIAL COMMUNICATION PROTOCOL USING A SERIAL DATA LINE (SDA) AND A SERIAL CLOCK LINE (SCL). THE PROTOCOL SUPPORTS MULTIPLE TARGET DEVICES ON A COMMUNICATION BUS AND CAN ALSO SUPPORT MULTIPLE CONTROLLERS THAT SEND AND RECEIVE COMMANDS AND DATA.



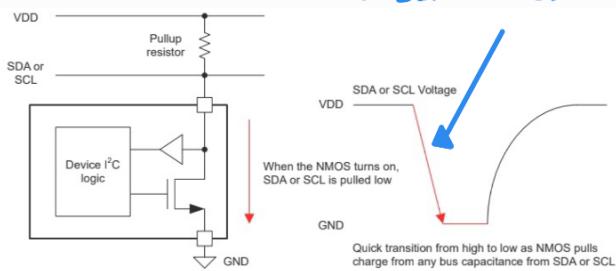
I<sup>2</sup>C IS HALF-DUPLEX COMMUNICATION WHERE ONLY A SINGLE CONTROLLER OR TARGET DEVICE IS SENDING DATA ON THE BUS AT A TIME.



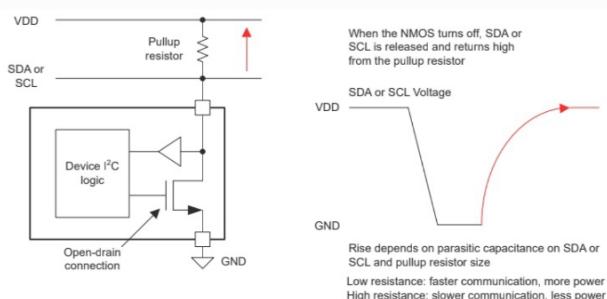
AN I<sup>2</sup>C CONTROLLER DEVICE SIGNS AND STOPS COMMUNICATION, WHICH REMOVE THE POTENTIAL PROBLEM OF BUS CONVENTION

# OPEN-DRAIN

THE SPEED OF THE TRANSITION IS DETERMINED BY THE NMOS DRIVE STRENGTH AND ANY BUS CAPACITANCE ON SDA OR SCL



IN THIS CASE THE NMOS IS ON, SO THE DEVICE PULLS CURRENT THROUGH THE RESISTOR TO GROUND. THIS PULLS THE OPEN-DRAIN LINE LOW.

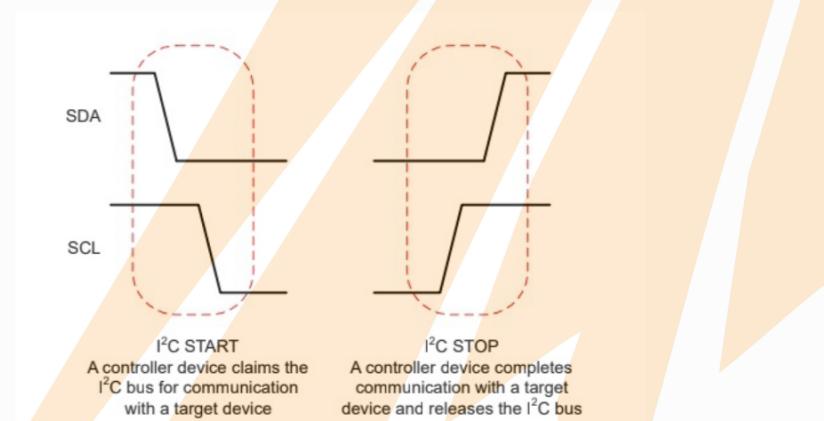


WHEN THE NMOS IS OFF, THE DEVICE STOPS PULLING CURRENT AND THE PULLUP RESISTORS PULLS SDA OR SCL LINE TO VDD.

# Protocol

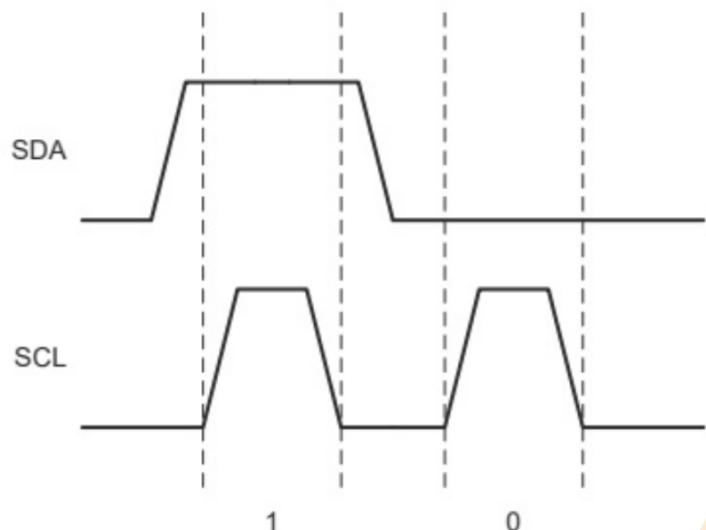
## • START AND STOP

THE COMMUNICATION IS INITIATED FROM THE CONTROLLER DEVICE WITH AN I<sup>2</sup>C START CONDITION. IF THE BUS IS OPEN THEN AN I<sup>2</sup>C CONTROLLER CLAIMS THE BUS FOR COMMUNICATION BY SENDING AN I<sup>2</sup>C START. TO DO THIS THE CONTROLLER DEVICE FIRST PULLS THE SDA LOW AND THEN PULLS THE SCL LOW.



INITIALLY BOTH SDA AND SCL ARE HIGH, INDICATING THAT THE LINE IS FREE. ONCE THE MASTER DECIDE TO COMMUNICATE IT TURNS LOW SDA. SINCE WE ARE WORKING WITH A FIXED FREQUENCY BASE ON OUR SPEED COMMUNICATION, IF SDA STAY LOW ENOUGH, IT WILL TRIGGER THE SCL TO BE LOW TOO, AND THE START CONDITION IS SENT

# LOGICAL ONE AND ZEROS



To let the slave to acquire data we must say it when to read them from SDA.

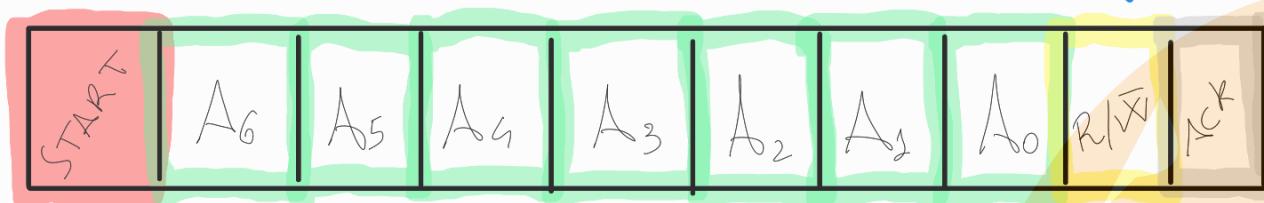
If we want transmit a 1, the master will turn high the SDA line for a certain amount of time given by the speed configuration we are using.

Also after a certain amount of time the SDA is high, the SCL is activated so the slave can read the most stable part of the signal. This is done to be sure the bit is correctly read.

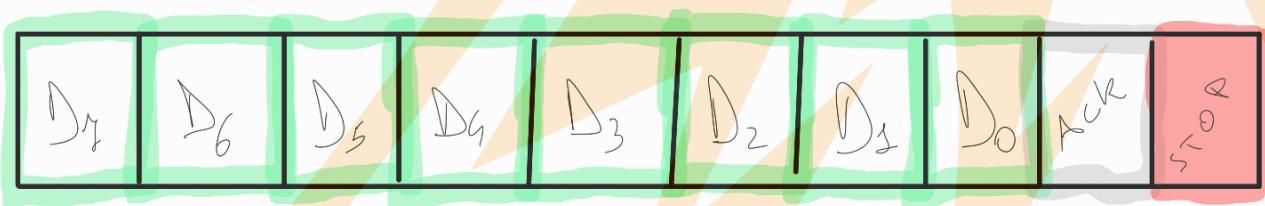
Since we know before communication our work-frequency, we only need to generate a pulse with that frequency for our SCL.

# COMMUNICATION FRAMES

THE SLAVE IS SET TO READ OR WRITE MODE



WHEN THE MASTER STARTS TO COMMUNICATE, THE FIRST GROUP OF BYTE TRANSMITTED IS THE ADDRESS OF THE DEVICE. EACH A<sub>i</sub> BIT IS SENT ONE AT A TIME AND APPEARS AS A PULSE ON THE OSCILLOSCOPE. EACH BIT IS READ FROM EACH SLAVE, WHEN 8 BITS ARE READ THE SLAVE CHECKS IF THE ADDRESS IS HIS. IF IS HIS BYTE AND SEND ALSO AN ACK, BASICALLY IT SETS HIGH THE SDA LINE SO THE MASTER KNOWS THAT THE COMMUNICATION IS ESTABLISHED.



THE SECOND GROUP OF BYTES MAY BE DIFFERENT THINGS: THE ADDRESS OF THE REGISTER, THE DATA OR OTHER.

ONCE EVERYTHING IS TRANSMITTED A STOP SIGNAL IS PASSED IN THE LINES: SDA IS HIGH WHILE SCL IS LOW.