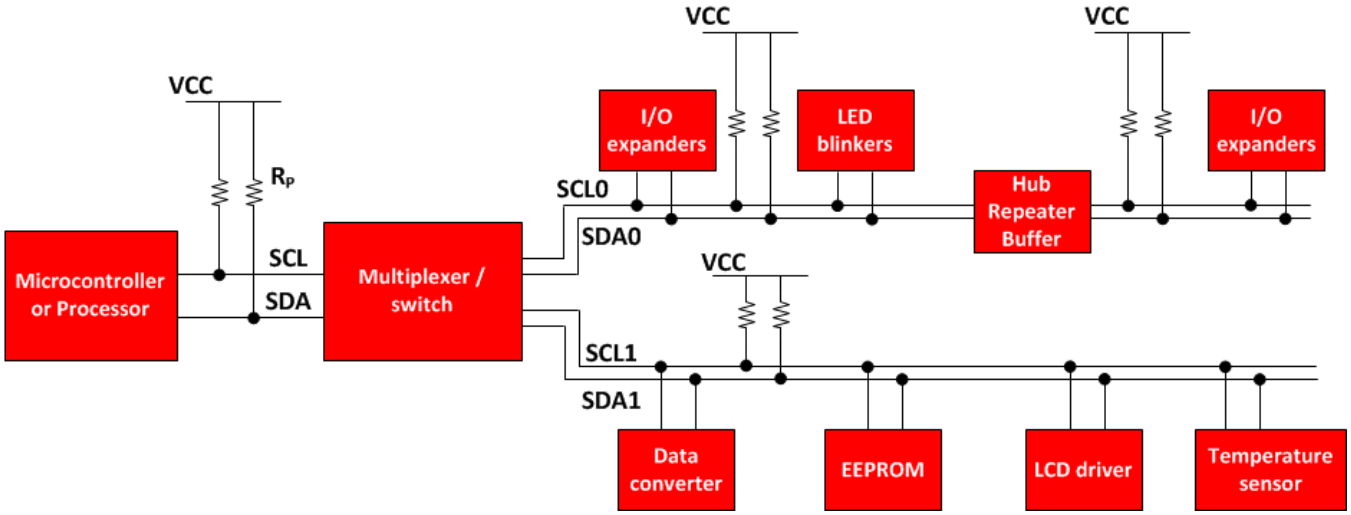


Figure 1. Application Example Showing I2C Communication Between the Different IC's on a System and With Pullup Resistors on I2C Bus

2 Pullup Resistor Calculation

A strong pullup (small resistor) prevents the I2C pin on an IC from being able to drive low. The V_{OL} level that can be read as a valid logical low by the input buffers of an IC determines the minimum pullup resistance [R_P(min)]. R_P(min) is a function of V_{CC}, V_{OL} (max), and I_{OL}:

$$R_P(\text{min}) = \frac{(V_{CC} - V_{OL}(\text{max}))}{I_{OL}} \quad (1)$$

The maximum pullup resistance is limited by the bus capacitance (C_b) due to I2C standard rise time specifications. If the pullup resistor value is too high, the I2C line may not rise to a logical high before it is pulled low. The response of an RC circuit to a voltage step of amplitude V_{CC}, starting at time t = 0 is characterized by time constant RC. The voltage waveform can be written as:

$$V(t) = V_{CC} \times \left(1 - e^{\frac{-t}{RC}}\right) \quad (2)$$

For $V_{IH} = 0.7 \times V_{CC}$:

$$V_{IH} = 0.7 \times V_{CC} = V_{CC} \times \left(1 - e^{\frac{-t1}{Rp \times C_b}}\right) \quad (3)$$

For $V_{IL} = 0.3 \times V_{CC}$:

$$V_{IL} = 0.3 \times V_{CC} = V_{CC} \times \left(1 - e^{\frac{-t2}{Rp \times C_b}}\right) \quad (4)$$

The rise time for the I2C bus can be written as:

$$t_r = t_2 - t_1 = 0.8473 \times R_p \times C_b \quad (5)$$

The maximum pullup resistance is a function of the maximum rise time (t_r):

$$R_p(\text{max}) = \frac{t_r}{(0.8473 \times C_b)} \quad (6)$$

where parametrics from I2C specifications are listed in Table 1.

9 Digital Interface

9.1 I²C and SPI (MPU-6000 only) Serial Interfaces

The internal registers and memory of the MPU-6000/MPU-6050 can be accessed using either I²C at 400 kHz or SPI at 1MHz (MPU-6000 only). SPI operates in four-wire mode.

Serial Interface

Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
8	Y		/CS	SPI chip select (0=SPI enable)
8		Y	VLOGIC	Digital I/O supply voltage. VLOGIC must be ≤ VDD at all times.
9	Y		AD0 / SDO	I²C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I²C Slave Address LSB
23	Y		SCL / SCLK	I²C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I²C serial clock
24	Y		SDA / SDI	I²C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I²C serial data

Note:

To prevent switching into I²C mode when using SPI (MPU-6000), the I²C interface should be disabled by setting the *I2C_IF_DIS* configuration bit. Setting this bit should be performed immediately after waiting for the time specified by the "Start-Up Time for Register Read/Write" in Section 6.3.

For further information regarding the *I2C_IF_DIS* bit, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.

9.2 I²C Interface

I²C is a two-wire interface comprised of the signals serial data (SDA) and serial clock (SCL). In general, the lines are open-drain and bi-directional. In a generalized I²C interface implementation, attached devices can be a master or a slave. The master device puts the slave address on the bus, and the slave device with the matching address acknowledges the master.

The MPU-60X0 always operates as a slave device when communicating to the system processor, which thus acts as the master. SDA and SCL lines typically need pull-up resistors to VDD. The maximum bus speed is 400 kHz.

The slave address of the MPU-60X0 is b110100X which is 7 bits long. The LSB bit of the 7 bit address is determined by the logic level on pin AD0. This allows two MPU-60X0s to be connected to the same I²C bus. When used in this configuration, the address of the one of the devices should be b1101000 (pin AD0 is logic low) and the address of the other should be b1101001 (pin AD0 is logic high).

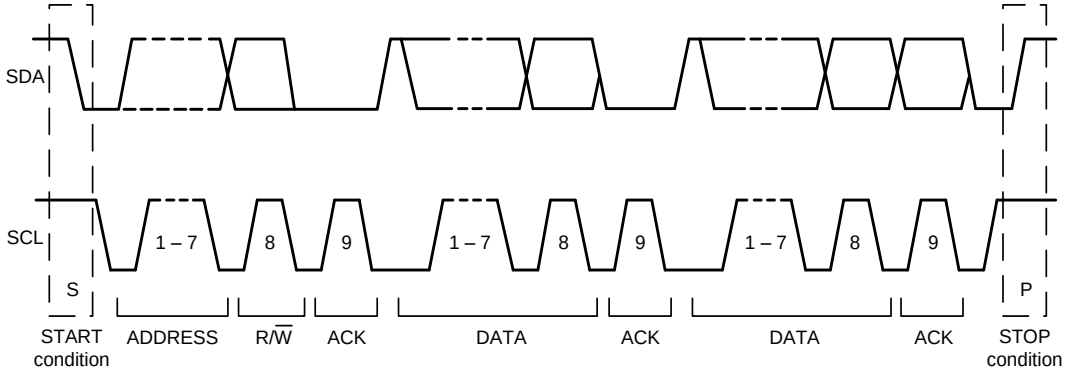
9.3 I²C Communications Protocol

START (S) and STOP (P) Conditions

Communication on the I²C bus starts when the master puts the START condition (S) on the bus, which is defined as a HIGH-to-LOW transition of the SDA line while SCL line is HIGH (see figure below). The bus is considered to be busy until the master puts a STOP condition (P) on the bus, which is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH (see figure below).

Communications

After beginning communications with the START condition (S), the master sends a 7-bit slave address followed by an 8th bit, the read/write bit. The read/write bit indicates whether the master is receiving data from or is writing to the slave device. Then, the master releases the SDA line and waits for the acknowledge signal (ACK) from the slave device. Each byte transferred must be followed by an acknowledge bit. To acknowledge, the slave device pulls the SDA line LOW and keeps it LOW for the high period of the SCL line. Data transmission is always terminated by the master with a STOP condition (P), thus freeing the communications line. However, the master can generate a repeated START condition (Sr), and address another slave without first generating a STOP condition (P). A LOW to HIGH transition on the SDA line while SCL is HIGH defines the stop condition. All SDA changes should take place when SCL is low, with the exception of start and stop conditions.



Complete I²C Data Transfer

To write the internal MPU-60X0 registers, the master transmits the start condition (S), followed by the I²C address and the write bit (0). At the 9th clock cycle (when the clock is high), the MPU-60X0 acknowledges the transfer. Then the master puts the register address (RA) on the bus. After the MPU-60X0 acknowledges the reception of the register address, the master puts the register data onto the bus. This is followed by the ACK signal, and data transfer may be concluded by the stop condition (P). To write multiple bytes after the last ACK signal, the master can continue outputting data rather than transmitting a stop signal. In this case, the MPU-60X0 automatically increments the register address and loads the data to the appropriate register. The following figures show single and two-byte write sequences.

Single-Byte Write Sequence

Master	S	AD+W		RA		DATA		P
Slave			ACK		ACK		ACK	

Burst Write Sequence

Master	S	AD+W		RA		DATA		DATA		P
Slave			ACK		ACK		ACK		ACK	

To read the internal MPU-60X0 registers, the master sends a start condition, followed by the I^2C address and a write bit, and then the register address that is going to be read. Upon receiving the ACK signal from the MPU-60X0, the master transmits a start signal followed by the slave address and read bit. As a result, the MPU-60X0 sends an ACK signal and the data. The communication ends with a not acknowledge (NACK) signal and a stop bit from master. The NACK condition is defined such that the SDA line remains high at the 9th clock cycle. The following figures show single and two-byte read sequences.

Single-Byte Read Sequence

Master	S	AD+W		RA		S	AD+R			NACK	P
Slave			ACK		ACK			ACK	DATA		

Burst Read Sequence

Master	S	AD+W		RA		S	AD+R			ACK		NACK	P
Slave			ACK		ACK			ACK	DATA		DATA		

9.4 I^2C Terms

Signal	Description
S	Start Condition: SDA goes from high to low while SCL is high
AD	Slave I^2C address
W	Write bit (0)
R	Read bit (1)
ACK	Acknowledge: SDA line is low while the SCL line is high at the 9th clock cycle
NACK	Not-Acknowledge: SDA line stays high at the 9th clock cycle
RA	MPU-60X0 internal register address
DATA	Transmit or received data
P	Stop condition: SDA going from low to high while SCL is high