**UART – Universal Asynchronous Receiver Transmitter**

* 5 to 9 data bits
* 1 start bit
* 0 to 1 parity bit
* 1 or 2 stop bits
* Asynchronous because communicating devices don’t share common clock
* Start bit used to let the device know that communication is about to start so that receiving device can synchronise its clock with the incoming data to receive it properly.
* Standard baud rates include: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
* 1 to 1 communication
* Least significant bit gets transferred first
* Requires 1 line for transmitting and one for receiving.
* Consequences of baud rate mismatch

<https://learn.sparkfun.com/tutorials/serial-communication/all>

<http://www.circuitbasics.com/basics-uart-communication/>

**SPI – Serial Peripheral Interface**

* Supports transmission of 8 or 16-bit data
* For operating modes: CPHA = CPOL = 0, CPHA = CPOL = 1, CPHA = 0 & CPOL = 1, CPHA = 1 & CPOL = 0.
* 4 Lines: CLK, MISO, MOSI and CS
* Data sent MSB first
* Clock signal is sent along with data
* CS lowered first before data is sent
* At high frequencies should not be used over long distances otherwise will give errors

<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all>

**IMU**

* Accelerometer, gyroscope and magnetometer theory
* Getting angles from the three theory
* Gyroscope noise, gimbal lock
* sensor accel, gyro and magnetometer offsets and noise and cross-axis sensitivity

**Complementary Filter**

* Complementary filter equation
* Theory

<https://www.pieter-jan.com/node/11>

<https://www.researchgate.net/publication/308850497_MEMS_based_IMU_for_tilting_measurement_Comparison_of_complementary_and_kalman_filter_based_data_fusion/download>

**Direct Memory Access**

* Theory and benefits

**Bluetooth Wireless**

**FPGA**

* Synchronisation is required so that metastability does not occur
* Controller Datapath architecture

**Euler Angles**

* Gimbal lock
* Theory

<https://en.wikipedia.org/wiki/Euler_angles>

<https://en.wikipedia.org/wiki/Gimbal_lock>

**UART – Universal Asynchronous Receiver Transmitter**

The UART communication protocol will be used to write data to the to the Bluetooth modules for wireless transmission. The protocol operates on 1 to 1 basis, meaning that the transaction can only happen between two devices at a time. This protocol is called asynchronous because the two sides of the transmission do not share a common clock and rely on their own clocks to synchronise themselves with the transmission of data in order to receive the information. The benefit of this is that the protocol only requires two lines to transmit data; one for transmitting and one for receiving data. The protocol consists of the following:

**Start Bit** – This bit is transmitted first to communicate with the other device that a transaction is about to begin. This bit is always 0 as the IDLE state of the communication line is of logic 1.

**Data Bits** – The data bits are transmitted after the start bit and the least significant bit of the data is transmitted first. The number of the data bits can range from 5 to 9.

**Parity Bit** – This bit is optional and is used as a low level error checking method. The parity can be odd or even. In order to produce the parity bit the data bits are added and the evenness of the data whether the bit is set or not. If parity is set to even and the number of 1’s in the data was odd, the parity bit would be set to 1. If the number of 1’s was even the parity bit would be set to 0. The opposite would happen if the parity was set to odd; if number of 1’s is odd the parity bit would be 0 and when the number of 1’s was even, the parity would be set to 1.

**Stop bits** – The stop bit is transmitted last and is a transition to the IDLE state which is 1 and therefore the stop bit is always 1. The stop bit can be either one or two bits wide depending on configuration.

The speed of transmission is determined by the baud rate which measures the amount of bits transmitted per second. The standard baud rates in devices are: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200. The communicating devices need to be configured identically in order to avoid data mismatch which will cause the data to be received incorrectly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UART Data Frame** | Start | Data | Parity | Stop |
| **Size** | 1 | 5-9 | 0-1 | 1-2 |

**SPI – Serial Peripheral Interface**

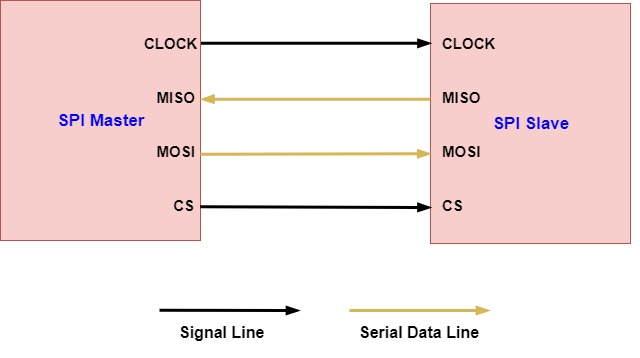
The Serial peripheral interface will be used for communication between internal measurement units and FPGA and also for communication between the FPGA and STM32L432KC. Unlike the UART protocol, SPI provides a synchronous solution to communication by having a separate line for the clock signal that is sent along with the data. The serial peripheral interface also possesses the ability to communicate with more than one slave device by using an additional chip select line for each device connected to the interface. The Serial peripheral interface therefore requires four lines to enable communication between devices. These lines are:

**CLOCK** – The clock signal is sent to slave device from master using this line.

**MOSI** – The Master Out Slave In line is used to send data from the master to the slave.

**MISO** – The Master In Slave Our line is used to send data from the slave device to the master

**CS** – The Chip Select line, also known as Slave Select (SS or SSEL) is used by the master to select the slave device with which the master wishes to communicate with.



Before the data is transmitted the SPI master lowers the chip select line to notify the slave device that a transmission is about to begin. The master device then provides the clock signal to which the data being sent is synchronised. once all the data bits have been sent, the master pulls the chip select high to end the transaction. In contrast to the UART protocol, the data is sent most significant bit first and The data transmitted can only be configured to be 8 or 16 bits wide. There are also four communication modes in SPI which are determined by the clock polarity (CPOL) and the clock phase (CPHA). The clock polarity determines the edge on which the serial peripheral interface transitions to the idle state. The clock phase determines which clock edge is used to place data on the line and which clock edge is used to capture the data on the receiving end.

**CPOL = 0 & CPHA = 0** – The clock starts on the rising edge and goes idle on the falling edge. Data is placed on the line on falling edges and is captured on rising edges of the clock. In this mode, the first bit has to be present on the line before the transmission starts.

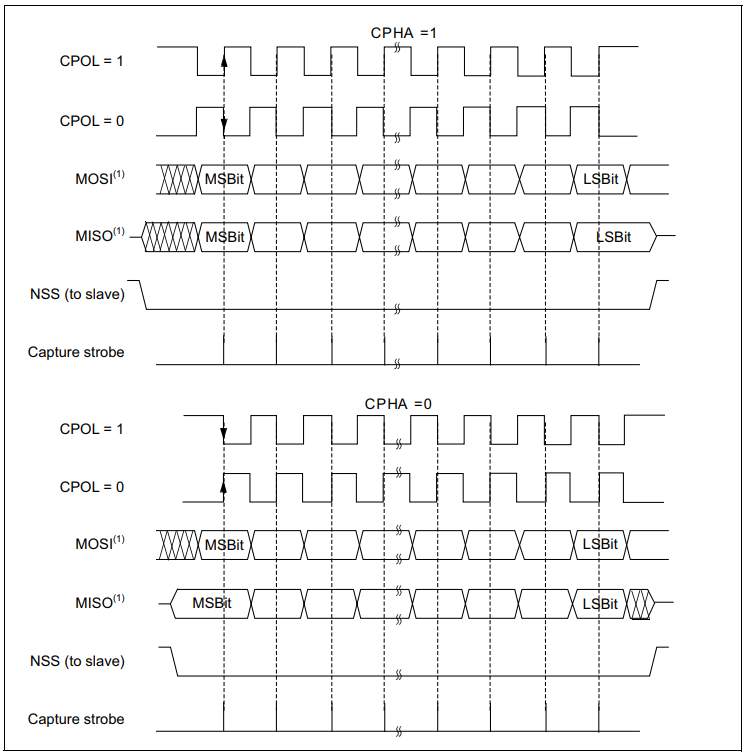
**CPOL = 1 & CPHA = 0** – The clock starts on the falling edge and goes idle on the rising edge. Data is placed on the line on rising edges and captured on falling edges of the clock.

**CPOL = 0 & CPHA = 1** – The clock starts on the rising edge and goes idle on the falling edge. Data is placed on the line on rising edges and is captured on falling edges of the clock.

**CPOL = 1 & CPHA = 1** – The clock starts on the falling edge and goes idle on the rising edge. Data is placed on the line on falling edges and is captured on rising edges of the clock.

The communicating devices must be configured identically in order for the transaction to be successful.

The communication protocol is capable of working at much higher frequencies (millions of bits per second) compared to UART. Due to higher frequency of operation the serial peripheral interface is not suited for long range communications as capacitive effects of the wire become significant with increasing frequency and distance causing the signal to distort and being interpreted by the receiving device incorrectly.



**IMU – Internal Measurement Unit**

The internal measurement unit also know as an IMU, typically consists of 3-axis accelerometer, gyroscope and magnetometer. Many of the low-end internal measurement units use MEMS (Micro-Electro-Mechanical-Systems) accelerometers and gyroscopes and employ magnetometers which rely on the hall-effect to produce an output.

MEMS technology can be defined as miniaturised mechanical and electro-mechanical elements that are made using the techniques of microfabrication. The MEMS devices can vary from simple structures that have no moving parts to complex electromechanical systems that contain multiple moving elements under the control of integrated microelectronics. Some of these complex systems, such as the microsensors and microactuators are categorised as “transducers” which are defined as devices that convert energy from one form to another. A device such as a microsensor typically converts mechanical energy into an electrical signal.

<https://www.mems-exchange.org/MEMS/what-is.html>

***MEMS Accelerometer***

MEMS accelerometers are typically composed of movable proof mass with plates that are attached through a mechanical suspension system, and fixed outer plates, as shown in **Figure N**. The movable plates and fixed outer plates represent capacitors. The proof mass can only move up or down [1] causing the movable plates to shift thereby changing the capacitance C1 and C2. The output voltage of the system is proportional to the acceleration (which causes the movable plates to shift) felt by the proof mass which can be measured by the changes in capacitance C1 and C2 which acts as a voltage divider where is the output voltage and is the input voltage (Equation 1).

Eqn 1: Equation representing the output voltage obtained from MEMS accelerometer

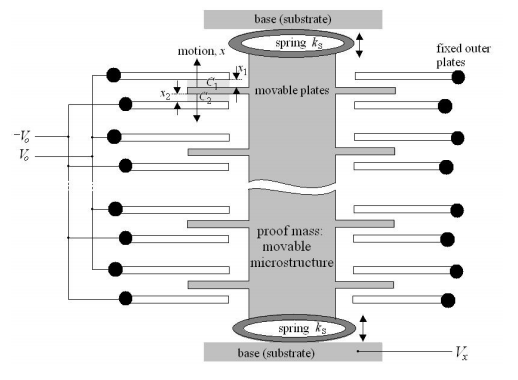


Figure n: Structure of MEMS accelerometer.

<http://mafija.fmf.uni-lj.si/seminar/files/2007_2008/MEMS_accelerometers-koncna.pdf>

***MEMS Gyroscope***

MEMS gyroscopes utilise the Coriolis force to detect angular velocities. The Coriolis effect states that a moving object that is subjected to a rotational force will experience a force perpendicular to the axis of rotation.

A common design of the MEMS gyroscope is the tuning-fork gyroscope. The tuning-fork Gyroscope contains a pair of proof masses that are made to oscillate with capacitive plates. The Gyroscope also consists of capacitive plates that are fixed in place besides the oscillating proof masses. When the structure is rotated, the Coriolis force induces oscillations in the proof masses at right angles to the axis of rotation. This causes the space between the capacitive plates to change while in motion. Due to the change in distance between the capacitive plates, the potential difference between the capacitive plates also changes.

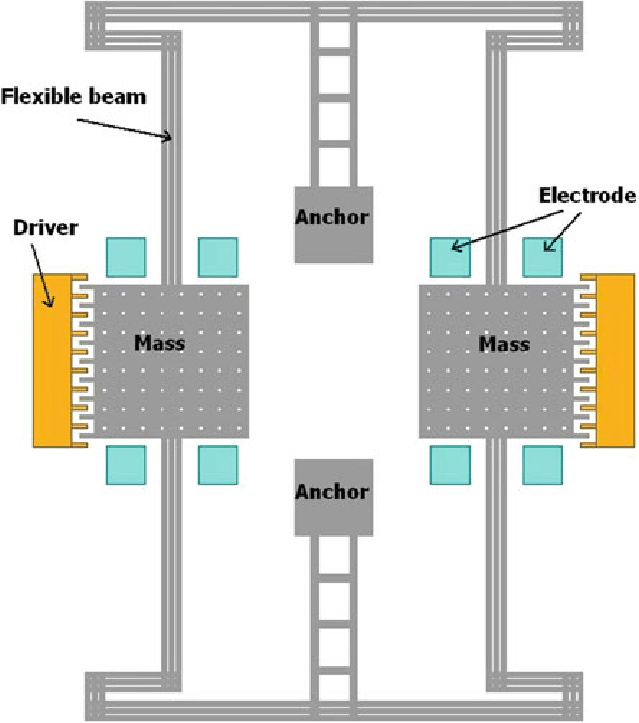
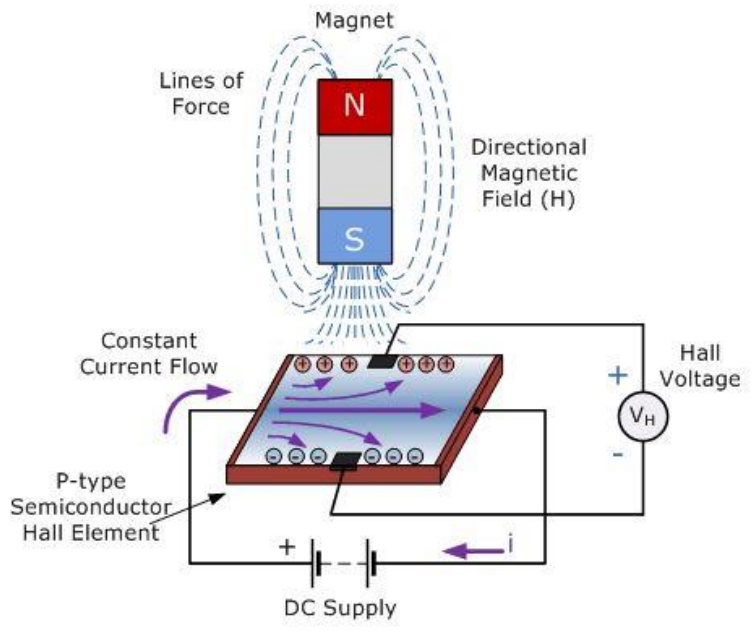


Figure n: Construction of MEMS gyroscope.

***Magnetometer***

The magnetometer uses the hall effect to measure earth’s magnetic field, or any magnetic field present. When a current is set to flow through a conductive plate, the electrons will travel straight to ground. If the plate is subjected to a magnetic field however, the electrons would be deflected, due to the Lorentz force experienced, to one side of the plate before flowing back to ground.

Because the electrons are deflected to one side, two regions in the plate are created, one where the plate is negatively charged and one where the plate is positively charged. Due to this a potential difference develops across the plate, the magnitude of which can be measured. The stronger the magnetic field the plate is exposed to, the higher the potential difference will be measured across the plate. This is called the Hall Effect. Magnetometers utilise this effect in order to output voltage based on the magnetic field present [6].

Figure n: Electrons deflected to one side of the plate in the presence of a magnetic field. Source [2].

**IMU Errors**

***Bias Error***

Due physical differences during manufacture and wear to degradation over time, each internal measurement unit will exhibit unique amounts of bias error in its measurements. The bias error is defined as the difference in the value obtained and the expected value. For example, the gyroscope at rest should return a value of zero (assuming no noise error) but will instead return a non-zero value despite remaining stationary.

***Noise Error***

Noise error occurs randomly and is therefore described as a stochastic process. This error is small but will accumulate during the integration of angular velocity from the gyroscope causing the orientation values to eventually drift away from the correct value. The noise error is usually minimised through statistical techniques such as noise averaging.

***Gyroscope Drift***

The gyroscope drift is caused by the noise that is inherent in the system. For the gyroscope to be useful in determining the orientation of the arm, the data obtained from the device needs to be continually integrated with respect to time. Because this process is incremental, the noise error will accumulate over time and cause the orientation value to drift away from the correct value.

***Sensor Non-orthogonality***

When three gyroscopes, accelerometers and magnetometers are produced, they are intended to be mounted orthogonally in the x, y and z axes. In reality the mountings are not always perfect and as a result the sensors are not exactly at 90 degrees relative to each other [3]. This leads to correlation between sensors. What this means is that when an accelerometer, for example, is measuring gravity (assuming the sensor is placed at right angles to gravity vector) along the gravity vector, the other two accelerometers will measure this force to a certain extent.

**Calculating Angles**

***Accelerometer***

The angle calculations using accelerometer readings rely on the gravity vector to obtain the results. This vector always points downward along the z-axis of the accelerometer when no tilt is applied to the device and is equal to 1g. When the internal measurement unit is rotated about the x or y axis, a component of the gravity vector can be felt by the x or y or both x and y axes (depending on how the device was rotated) as the axes are no longer orthogonal relative to the gravity vector. The sum of components of the gravity vector in the x, y and z is always 1g as shown by equation 2.

Eqn 2: Sum of gravity components in x, y and z axes is always equal to the gravity vector.

To calculate the tilt angles in the x and y axes, equations 3 and 4 are utilised. It is not possible to calculate the rotation angle in the z-axis due to the fact these rotations do not change the gravity vector component in the z-axis and therefore the rotation cannot be resolved.

Eqn 3: Tilt angle in the x-axis.

Eqn 4: Tilt angle in the y-axis.

***Gyroscope***

The angular position in the x, y and z axes can simply be calculated by integrating the angular velocity detected by the sensor with respect to time.

Eqn 5: Obtaining angular position by integrating angular velocity with respect to time [4].

In digital systems integration is the sum of all angular velocity samples, , multiplied by the sampling period .

Eqn 6: Digital method of angular velocity integration [4].

The accuracy of the integration depends on the sampling frequency of the system. If the changes in angle are faster than the sampling frequency of the system, the changes in angle will not be recorded. Therefore, to achieve highest accuracy possible the sampling frequency should be as high as necessary to maintain an adequate level of accuracy.

***Magnetometer***

Similar to how the gravity vector is used to calculate the tilt in the and y axes using the accelerometer, the magnetometer uses earth’s magnetic field to determine the magnetic field vector components in the x and z axes which can be used to calculate the rotation angle of the device in the z-axis. Using equation 7, the rotation angle in the z-axis can be calculated. points to the initial state of . The initial state is equal to zero if the device’s coordinate frame is aligned with the earth’s coordinate frame.

# **References**

[1] S. E. Lyshevski, Mems and Nems: systems, devices and structures (CRC Press LLC, USA, 2002)

[2] <https://blog.digilentinc.com/what-is-the-hall-effect/>

[3] <https://www.novatel.com/assets/Documents/Bulletins/APN064.pdf>

[4] Pieter-Jan Van de Maele. Getting the angular position from gyroscope data. September 6th, 2012. Available at: <https://www.pieter-jan.com/node/7>