The Testing *FS-i6X* & *FS-iA6B*

Radio control (RC) is the use of control signals transmitted by a radio to remotely control a device.

FS-i6X Transmitter (Tx) by Flysky is a 6-channel wireless transmitter at a frequency of 2.4GHZ using power less than 20dBm, and uses the protocol *Automatic Frequency Hopping Digital System* (AFHDS) allowing for bidirectional data transmission. The data transmission supports **pulse-width-modulation (PWM)** and pulse-position-modulation (PPM). The packet transmission supports the i-bus and the s-bus.[1]

FS-iA6B Receiver (Rx) also by Flysky is compatible with the ***FS-i6X*** (Tx). The receiver can be battery powered from 4.0 to 8.4 volts.[2]

The MCU used for testing the RC devices was the Arduino Uno. Its function ***pulseIn(pin, value)*** can use either the value ***HIGH*** or the value ***LOW*** on an Arduino pin. If the value used is ***HIGH***, the function ***pulseIn***() waits for the pin to go from ***LOW*** to ***HIGH*** and then starts timing, then waits for the pin to go ***LOW*** and then, it stops timing. The function returns the length of the pulse in *microseconds* (μs).[3]

A *standardizing* method is needed to determine whether it can be viable for measuring the output of the RC devices.

Step (1): Use Arduino Uno to develop a method for testing the RC devices.
Step(2): Setup both the ***FS-i6X*** Transmitter and the ***FS-iA6B*** Receiver and then test the right gimbal (joystick) to observe the receivers PWM output and functionality of the right joystick.

---

1. FCC ID: N4ZFLYSKYI6X (https://www.flysky-cn.co) Flysky Website and Google FCC id.
2. FCC ID: N4ZFLYSKYIA10 (https://www.flysky-cn.co) Flysky Website and Google FCC id.
3. Arduino Website (https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/)

The **frequency** - f is the number of cycles required to complete in 1 second.

The **period** - T is the time required for one complete cycle to pass a given point.

The frequency is inversely proportional to the time period:

$$f\,(Hz) = \frac{1\,cycle}{T\,(s)} = \frac{1}{T\,(s)}$$

Total PWM Period:

$$T_{Total} = T_{HIGH} + T_{LOW}$$

PWM Frequency:

$$f = \frac{1}{T_{Total}\,(\mu s)} \cdot \frac{1\,000\,000\,(\mu s)}{1\,(s)} = \frac{1\,000\,000}{T_{Total}\,(s)}$$

PWM Duty Cycle:

$$Duty\ Cycle = \frac{T_{HIGH}}{T_{Total}}$$

Review the code at Github by MageMCU.