A correct BLE stream

Advertising:

preamble: 10101010 (if the first bit of Access Address is 1)
preamble: 01010101 (if the first bit of Access Address is **0**)

LSB

Access Address: 0x8E89BED6 (for advertising it’s a fixed pattern)
→1000 1110 1000 1001 1011 1110 1101 0110

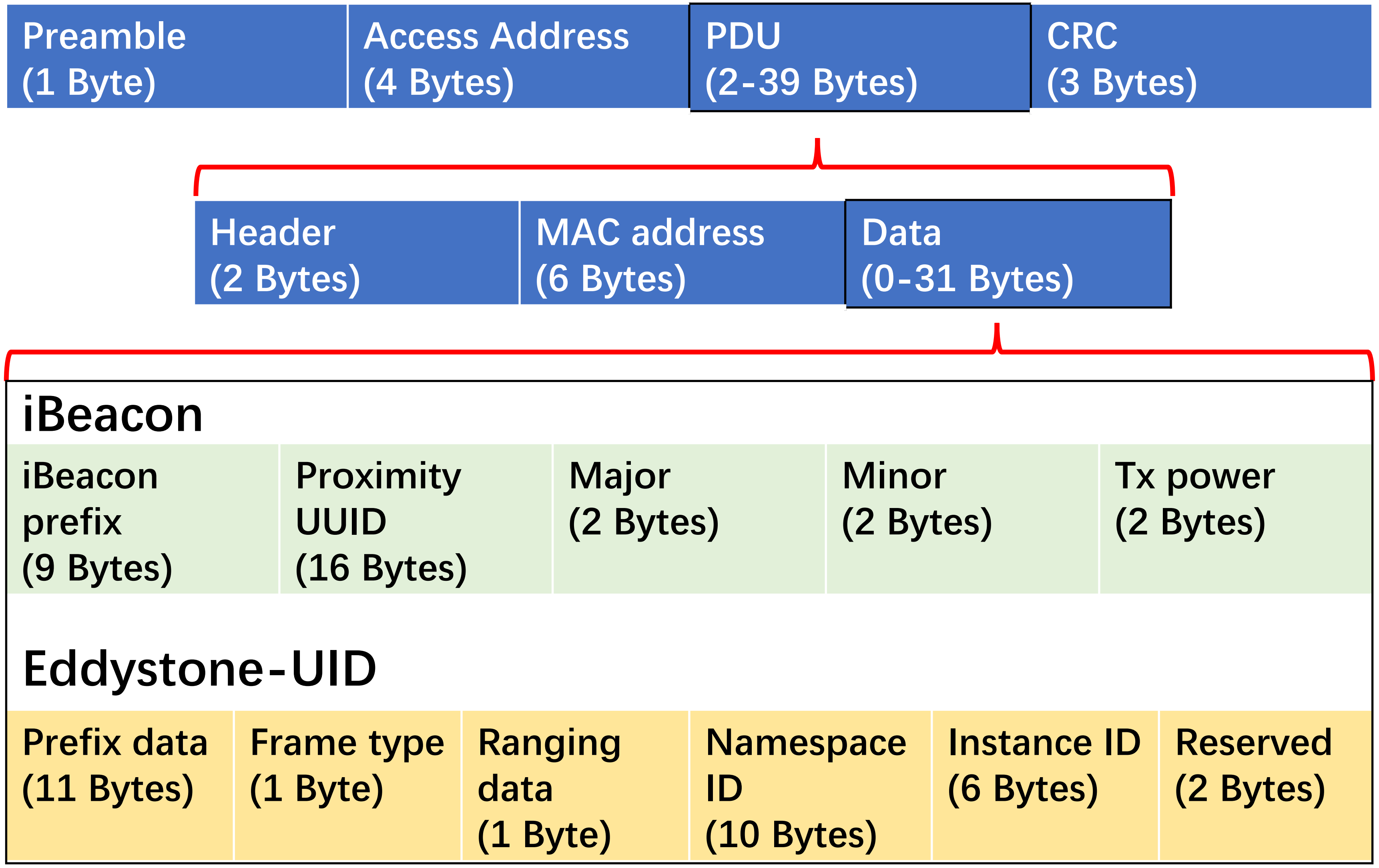
BLE Advertising packet format | BLE Data packet format (rfwireless-world.com)

So we need to find the sequence of **preamble** and **access address** after finished demodulation, to judge if it’s a BLE packet.

In theory, that is : 0101 0101 1000 1110 1000 1001 1011 1110 1101 0110

Packet formats of ibeacon and Eddystone-UID in advertising channel

In general:



Preamble (1 Byte)	Access Address (4 Bytes)	PDU (2-39 Bytes)	CRC (3 Bytes)

Header (2 Bytes)	MAC address (6 Bytes)	Data (0-31 Bytes)

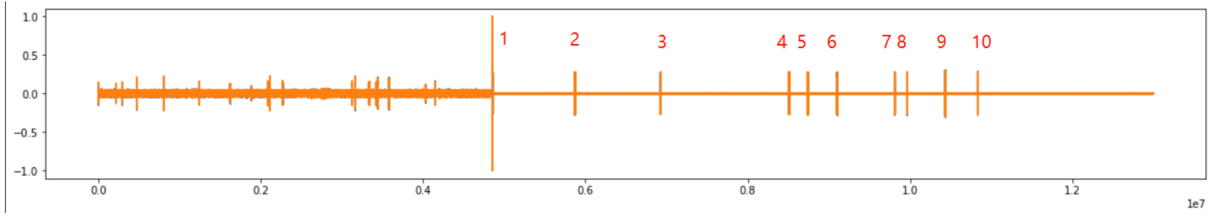
iBeacon

iBeacon prefix (9 Bytes)	Proximity UUID (16 Bytes)	Major (2 Bytes)	Minor (2 Bytes)	Tx power (2 Bytes)

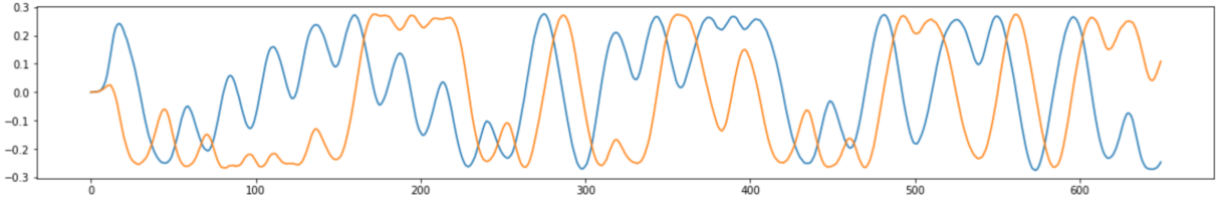
Eddystone-UID

Prefix data (11 Bytes)	Frame type (1 Byte)	Ranging data (1 Byte)	Namespace ID (10 Bytes)	Instance ID (6 Bytes)	Reserved (2 Bytes)

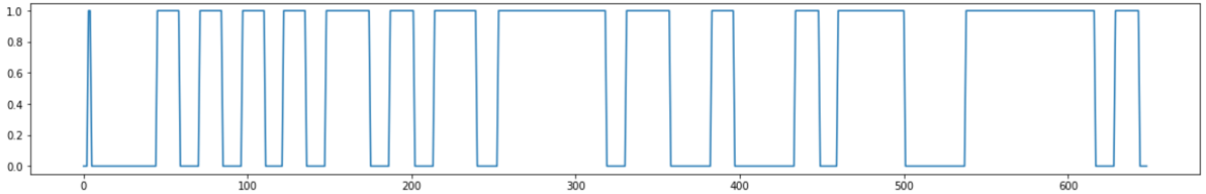
Eddystone-UID



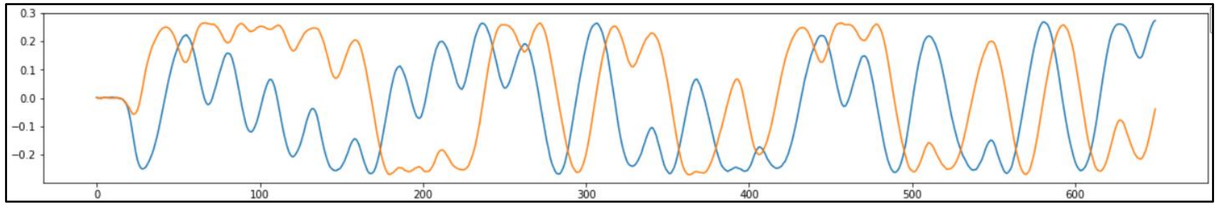
2. I/Q waveforms



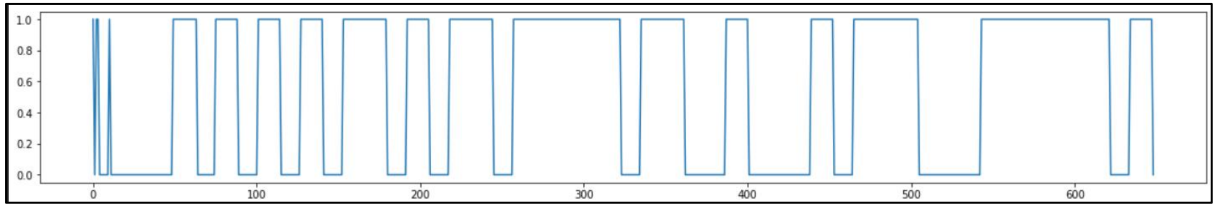
Before down sample



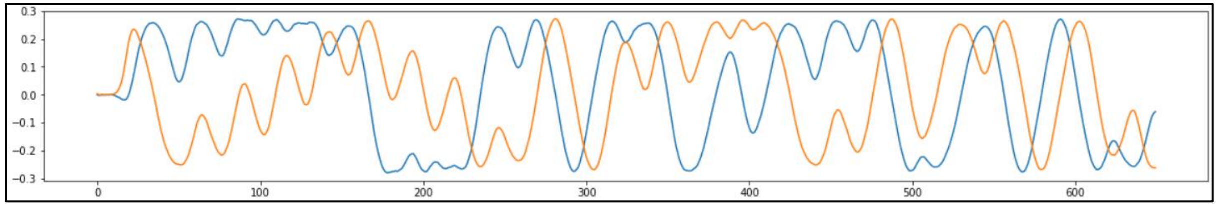
3. I/Q waveforms



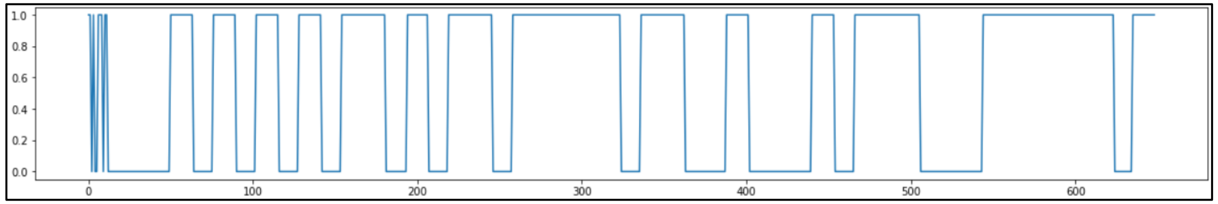
Before down sample



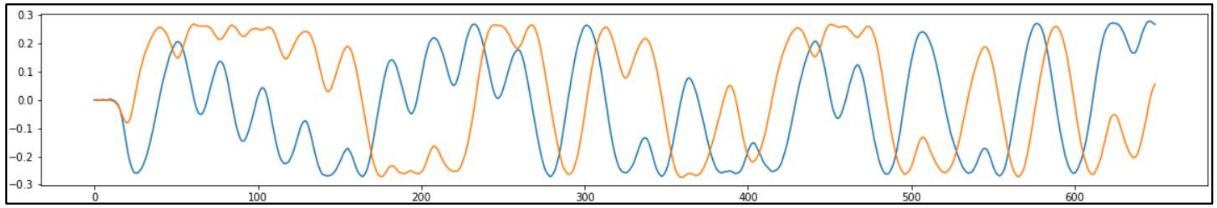
5. I/Q waveforms



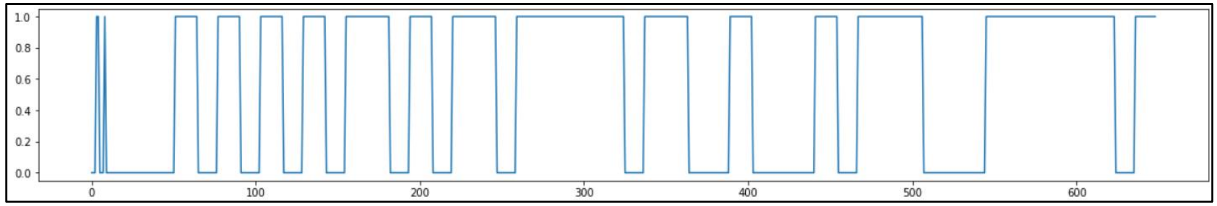
Before down sample



4. I/Q waveforms

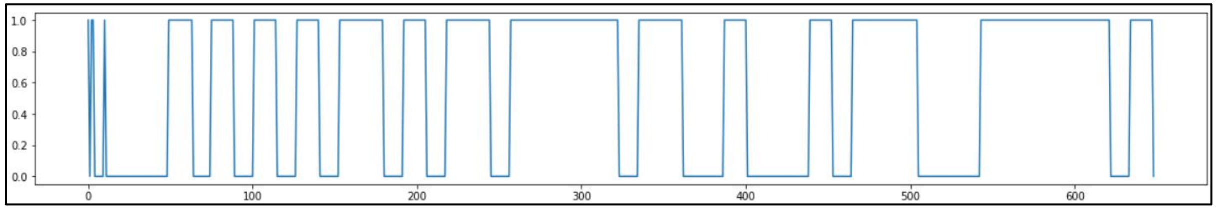


Before down sample

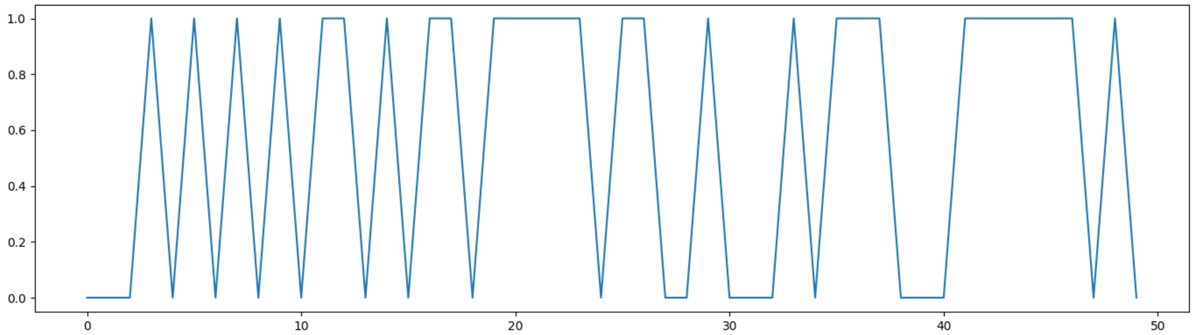


Eddystone-UID

Before down sample



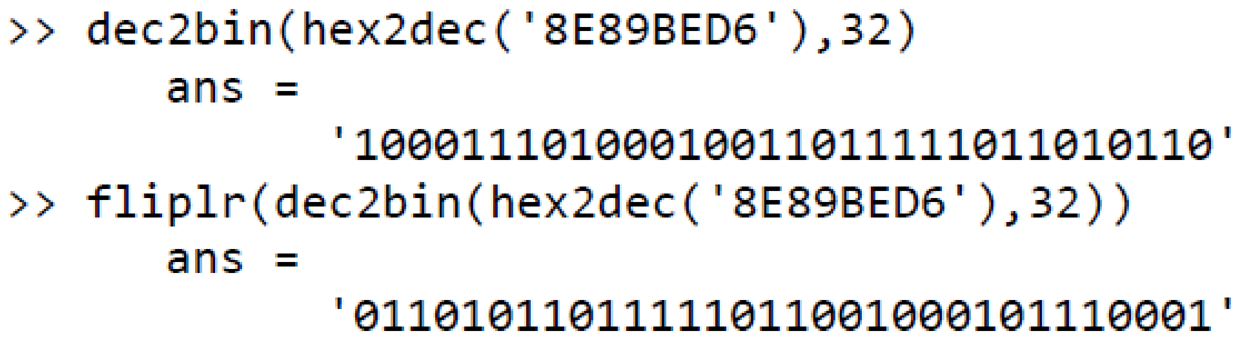
After down sample



After down sample: 00 01010101 0110 1011 0111 1101 1001 0001 0111 0001 1111 1010

	Preamble	Access Address	……
00	01010101	0110 1011 0111 1101 1001 0001 0111 0001	1111 1010
	0x55	0x6B7D9171	

Then



```
>> dec2bin(hex2dec('8E89BED6'),32)
    ans =
          '10001110100010011011111011010110'
>> fliplr(dec2bin(hex2dec('8E89BED6'),32))
    ans =
          '01101011011111011001000101110001'
```

So, actually

0x8E89BED6 == 1000 1110 1000 1001 1011 1110 1101 0110 original

0x6B7D9171 == 0110 1011 0111 1101 1001 0001 0111 0001 final