

# Hamster Manual

## 1. Hamster Technical Specification

	Spec	Detail
Size	35 x 35 x 30 mm	L x W x H
Weight	35 g	
Communication	Bluetooth 4.0 BLE	Supports Bluetooth 4.1 BLE
Required OS	Android 4.3 or above, iOS, OS X	Window OS requires USB dongle
Battery	Li-poly, 3.7 volt, 100mA PCM (Protection Circuit Module)	Operation: 1 hr, Stand by: 12 hr Recharging: 30 min
Display light	BLE connection (blue), recharging (red) left LED (RGB), right LED (RGB)	7 colors
Sampling rate	Sensors: max 50 Hz (20 msec) Actuators: max 50 Hz (20 msec)	10~50 Hz
Actuator	DC geared motor x 2	Planetary gear
Buzzer	1.00 ~ 165,000.00 Hz, monotone	Resolution: 0.01 Hz
Music notes	88 key, A3 ~ A7, 12 equal temperament	precision: +/-0.1 cent
IR Proximity	2 sensors (left/right), max distance 30 cm	50 Hz
Light detection	Front facing, 0~65,000 Lux	10 Hz
Line tracing	2 sensors (left/right)	50 Hz
Accelerometer	3 Axes (x, y, z)	Resolution setting: 2, 4, 8, 16G
Temperature	-40.0 ~ +87.5 deg (Celsius)	Resolution: 0.5 deg (Celsius)
External input*	2 ports: Analog, Digital	0 ~ 3.3 volt
External output*	2 ports: Analog, Digital, Servo, PWM	PWM 255 stages, 0~180 deg
Recharging	Micro-USB	Smart phone cable (5 pin)
BLE host/client	Connects up to 3 other BLE devices	Slower communication speed
Other	Battery voltage, Signal strength	
Power	Slider switch	Power on/off

- External I/O: Each port can use either input mode or output mode at a time. Note that it is possible to change I/O mode during operation.

## 2. Hamster advertising packet definition

Scanning: Advertising packet broadcasted by Hamster

```
[-] Hamster (0xFC11FF7E6ADD) (-62dBm)
  ... RSSI: -62dBm
  ... Address: FC11FF7E6ADD
  ... Address Type: Random
  ... Advertising Type: Connectable
  ... Bonded: False
  [-] Advertising Data
    ... CompleteLocalName: Hamster
    ... Appearance: 0x0402
    ... Flags: GeneralDiscoverable, BrEdrNotSupported
    ... ServicesCompleteListUuid16: 0xF138
  ... Scan Response Data
```

### 2.1 Product Name

Complete Local Name: Hamster

### 2.2 Bluetooth Address: 0xFC11FF7E6ADD is the unique product address

- Address Type: Random

### 2.3 Advertising Type: Connectable, Bonded: False

- connectible device, no pairing

### 2.4 Appearance:

- 16 bit ID number: used to differentiate products: Hamster: 0x0403 (PID+DID)

- First byte: Product ID(0x04) - product model

(if model numbers are the same, the control data are the same)

- Second byte: Device ID(0x03) - product version

(number for checking the product type without actual connection)

### 2.5 Flags:

- General Discoverable: broadcasting Advertising signal continuously until connecting to the host (smart devices/computer)

- BrEdrNotSupported: supports on Bluetooth 4.0 BLE

No registration or pairing required: can connect to any host without pairing process.

### 2.6 ServiceCompleteListUuid:

- 16 bit UUID: 0xF138.

- Among the devices satisfying conditions 2.1~2.5 above, connect the one with service ID 0xF138.

## 3. Hamster's BLE Primary Services Definition

### 3.1 Generic Access Service (0x1800)

- service defined in Bluetooth BLE

- Min/Max Connection Interval: 20 msec

### 3.2 Generic Attribute Service (0x1801)

- service defined in Bluetooth BLE

### 3.3 Custom Defined Service "Sensors"

UUID: 0x00009001-9C80-11E3-A5E2-0800200C9A66 communication with host



```

PrimaryService, Value: 00-18, Generic Access (0x1800)
├── CharacteristicDeclaration, Value: 04-03-00-00-2A, Properties: Read, Write, Characteristic UUID: 0x2A00
│   ├── DeviceName, Value: 48-61-8D-73-74-65-72, DeviceName: Hamster
│   ├── CharacteristicDeclaration, Value: 02-05-00-01-2A, Properties: Read, Characteristic UUID: 0x2A01
│   │   ├── Appearance, Value: 02-04, Appearance: 0x0402
│   │   └── CharacteristicDeclaration, Value: 02-07-00-04-2A, Properties: Read, Characteristic UUID: 0x2A04
│   └── SlavePreferredConnectionParameters, Value: 10-00-10-00-00-32-00, MinConnInterval: 0x0010, MaxConnInterval: 0x0010, SlaveLatency: 0x0000, SupervisionTimeoutMultiplier: 0x0032
├── PrimaryService, Value: 01-18, Generic Attribute (0x1801)
│   ├── PrimaryService, Value: 0A-18, Device Information (0x180A)
│   │   ├── CharacteristicDeclaration, Value: 02-0B-00-23-2A, Properties: Read, Characteristic UUID: 0x2A29
│   │   ├── Manufacturer Name String, Value: 52-6F-62-6F-6D-61-74-69-6F-6E
│   │   └── CharacteristicDeclaration, Value: 02-00-00-26-2A, Properties: Read, Characteristic UUID: 0x2A26
│   │       └── Firmware Revision String, Value: 31-2E-32
│   ├── PrimaryService, Value: 66-9A-0C-20-00-08-E2-A5-E3-11-80-9C-01-90-00-00, 0x00009001-9C80-11E3-A5E2-0800200C9A66
│   │   ├── CharacteristicDeclaration, Value: 14-10-00-66-9A-0C-20-00-08-E2-A5-E3-11-80-9C-0A-90-00-00, Properties: WriteWithoutResponse, Notify, Characteristic UUID: 0x0000900A-9C80-11E3-A5E2-0800200C9A66
│   │   │   ├── UUID: 0000900A-9C80-11E3-A5E2-0800200C9A66, Value: 00-00-10-00-1B-00-65-14-42-BF-12-BF-C7-7F-00-00-3C-F9-F8-00
│   │   │   ├── CharacteristicUserDescription, Value: 53-65-6E-73-6F-72-73, UserDescription: Sensors
│   │   │   └── ClientCharacteristicConfiguration, Value: 00-00, CharacteristicConfigurationBits: None (0x0000)
│   │   ├── PrimaryService, Value: 66-9A-0C-20-00-08-E2-A5-E3-11-80-9C-00-A0-00-00, 0x0000A000-9C80-11E3-A5E2-0800200C9A66
│   │   │   ├── CharacteristicDeclaration, Value: 04-15-00-66-9A-0C-20-00-08-E2-A5-E3-11-80-9C-06-A0-00-00, Properties: WriteWithoutResponse, Characteristic UUID: 0x0000A006-9C80-11E3-A5E2-0800200C9A66
│   │   │   │   ├── UUID: 0000A006-9C80-11E3-A5E2-0800200C9A66, Value: 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
│   │   │   │   ├── CharacteristicUserDescription, Value: 45-66-66-66-65-63-74-6F-72-73, UserDescription: Effectors
│   │   │   │   └── PrimaryService, Value: 23-01-BC-EA-5F-78-23-15-DE-EF-12-12-30-15-00-00, DFU (0x00001530-1212-EFDE-1523-785FEABCD123)
│   │   │   │       ├── CharacteristicDeclaration, Value: 04-19-00-23-01-BC-EA-5F-78-23-15-DE-EF-12-12-32-15-00-00, Properties: WriteWithoutResponse, Characteristic UUID: 0x00001532-1212-EFDE-1523-785FEABCD123
│   │   │   │       │   ├── DFU Packet (No values read)
│   │   │   │       │   ├── CharacteristicDeclaration, Value: 18-1B-00-23-01-BC-EA-5F-78-23-15-DE-EF-12-12-31-15-00-00, Properties: Write, Notify, Characteristic UUID: 0x00001531-1212-EFDE-1523-785FEABCD123
│   │   │   │       │   ├── DFU Control Point (No values read)
│   │   │   │       │   └── ClientCharacteristicConfiguration, Value: 00-00, CharacteristicConfigurationBits: None (0x0000)
│   │   │   └── CharacteristicDeclaration, Value: 02-1E-00-23-01-BC-EA-5F-78-23-15-DE-EF-12-12-34-15-00-00, Properties: Read, Characteristic UUID: 0x00001534-1212-EFDE-1523-785FEABCD123
│   │   │       ├── UUID: 00001534-1212-EFDE-1523-785FEABCD123, Value: 01-00

```

## 4. Sensors Service

### Sensors service: packet format definition (20 bytes)

	Details	Value from Robot	User converted value	Etc
0	<b>Version / Topology</b>	0 ~ 255	0 ~ 255	Ref.1
1	<b>Network ID</b>	0 ~ 255	0 ~ 255	Ref.2
2	<b>Command / Security</b>	0 ~ 255	0 ~ 255	Ref.3
3	<b>Signal Strength</b>	-128 ~ 0	-128 ~ 0 dBm	Ref.4
4	<b>Left Proximity</b>	0 ~ 255	0 ~ 255	Ref.5
5	<b>Right Proximity</b>	0 ~ 255	0 ~ 255	
6	<b>Left Floor</b>	0 ~ 255	0 ~ 255	Ref.6
7	<b>Right Floor</b>	0 ~ 255	0 ~ 255	
8	<b>Acc X High</b>	-32768 ~ 32767	-32768 ~ 32767	Ref.7
9	<b>Acc X Low</b>			
10	<b>Acc Y High</b>	-32768 ~ 32767	-32768 ~ 32767	
11	<b>Acc Y Low</b>			
12	<b>Acc Z High</b>	-32768 ~ 32767	-32768 ~ 32767	
13	<b>Acc Z Low</b>			
14	<b>Flag</b>			Ref.8
15	<b>Light High or Temperature</b>	0 ~ 65535 -128 ~ 127	0 ~ 65535 Lux -40 ~ 88 °C	Ref.9 Ref.10
16	<b>Light Low or Battery</b>	0 ~ 255	0 ~ 100 %	Ref.11
17	<b>Input A</b>	0~255	0 ~ 255 (0 ~ 3.3 V)	Ref.12
18	<b>Input B</b>			
19	<b>Line Tracer State</b>	0 ~ 255	0 ~ 255	Ref.13

#### Ref.1) Version, Topology

bit	7	6	5	4	3	2	1	0
value	Version				Topology			

Version: current version is 0

#### Topology Type

0	None (only connecting to a hub)
1	Daisy Chain
2	Star
3	Extended Star

#### Ref.2) Network ID

##### Topology 0: None

Network ID is 0x00

##### Topology 1: Daisy Chain(Line)

bit	7	6	5	4	3	2	1	0
value	Node 0 ~ 255							

### Topology 2: Star

bit	7	6	5	4	3	2	1	0
detail	Node 0 ~ 255							

### Topology 3: Extended Star(Cluster Tree)

bit	7	6	5	4	3	2	1	0
detail	Depth 4		Depth 3		Depth 2		Depth 1	

### Ref.3) Command, Security

bit	7	6	5	4	3	2	1	0
value	Command				Security			

Security == 0: not encrypted

Security != 0: encrypted (different methods depending on Security value)

```
int command = (packet[2] >> 4) & 0x0f;
```

```
int security = packet[2] & 0x0f;
```

### Command Types

1. Sensory packet: 0x1?

If (packet[2] >> 4) & 0x0f == 0x01, Hamster sensors packet

### Ref.4) Signal Strength

RSSI (Received Signal Strength Indication)

-- changes with direction and distance

-- operational up to -93dBm

-- optimal operation between -50dBm and -80dBm

### Ref.5) Left/Right Proximity

Infrared(IR) sensors located in the front

Name	Value (in hex)
Left Proximity	00 ~ FF (0 ~ 255, the closer, the bigger value)
Right Proximity	00 ~ FF (0 ~ 255, the closer, the bigger value)

\* Values depend on setting of Effectors' Proximity IR Current value (default setting: 10mA)

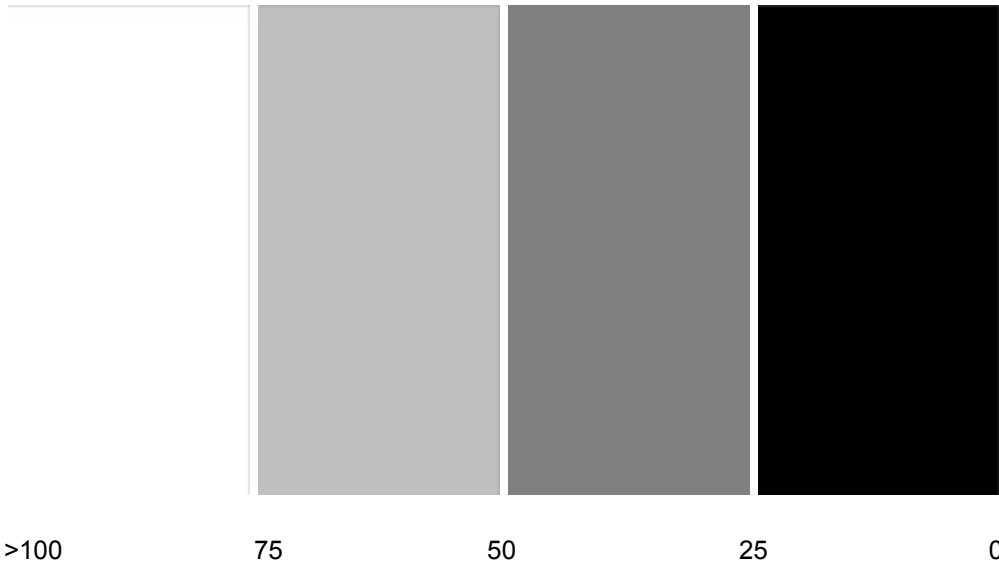
\* In case of the default setting (10mA), max detection ~10cm. For this case, 50 == ~5cm.  
(depending on materials)

### Ref.6) Left/Right Floor

Line sensors located on the bottom

Name	Value (in hex)
Left Floor	0 ~ 100 (100: white)
Right Floor	0 ~ 100 (100: white)

\* white balance used regular white paper for normal printers



### Ref.7) Acceleration

3D accelerometer (precision: 16 bit)

Name	Value(in hex)
Acc_X	-32768 ~ +32767(0x8000~0x7FFF)
Acc_Y	-32768 ~ +32767(0x8000~0x7FFF)
Acc_Z	-32768 ~ +32767(0x8000~0x7FFF)

\* Values depend on the setting of Effectors' G-Range value (default: 0x0 == 2g).

\* If G-Range is set for 2g, 2g == 32767(0x7FFF). In this case, the gravity (1g) == -16384 indicating negative Z-axis direction.

### Ref.8) Flag

0	if next 2 bytes are Light sensor values (high and low)
1	if next 2 bytes are Temperature and Battery

### Ref.9) Light

Light sensor located in the front

Name	Value (in hex)

Light(High, Low)	0000 ~ FFFF (0 ~ 65535), unit: lux
------------------	------------------------------------

### Ref.10) Temperature

Internal temperature in 2's complement (-40 ~ 87.5 Celsius with 24.0 as middle)

Formula) Temperature(in Celsius) = 24.0 + packet[15] / 2

ex) If packet[15] = 0xED(-19), 24.0 - 19/2 = 24.0 - 9.5 = 14.5 deg (in Celsius)

If packet[15] = 0x11(17), 24.0 + 17/2 = 24.0 + 8.5 = 32.5 deg (in Celsius)

(Caution) Temperature is the internal temperature of Hamster, not the room temperature.

### Ref.11) Battery

#### Voltage of the internal lithium-polymer battery

Formula) Battery(in Volt) = 2.0 + battery level / 100

ex) if packet[16] = 0x95(149), 2.0 + 149/100 = 2.0 + 1.49 = 3.49 volt

(Users use this value by converting to % and comparing to the battery graph)

(Caution) Hamster's DC regulated circuit uses 3.6 volt, voltage value = 3.6 +/- 0.1 volt

(Caution) If voltage < 3.5 volt, low voltage state (erratic sensor reading and effector writing, needs recharging)

(Caution) When fully charged, voltage = ~4.0 volt. The normal range is 3.7~3.8 volt.

### Ref.12) Input A/B

ADC mode) Analog to Digital Converter mode (Measuring analog voltage)

Active only if the Effectors' IO Mode value == 0

Formula) Volt = 3.3 \* ADC level / 255 (volt)

DI mode) Digital Input mode (Measuring digital input)

Active only if the Effectors' IO Mode value == 1

Formula) 1 if input voltage >= 0.5, 0 otherwise

### Ref.13) Line Tracer State

1)

bit	7	6	5	4	3	2	1	0
value	0	1	left wheel 0 ~ 11			right wheel 0 ~ 11		

If wheel value == 0, wheel is stopped.

If wheel is moving, wheel value = (real wheel value / 10) + 1

(Caution) If it is not line-tracing mode, wheel value == 0x00.

(Caution) If it is line-tracing mode and wheel is stopped, wheel value == 0x40.

### 2) Bar Code (Not implemented)

bit	7	6	5	4	3	2	1	0
value	1	0/1	Bar Code Number					

Detecting Bar Code value (6 bit)

(Caution) If the same values of bar code are detected consecutively, the 6 bit alternates 0/1.



## 5. Effectors Service

### Effectors service: packet format definition (20 bytes)

	Data	Value to Robot	User input value	Etc
0	<b>Version / Topology</b>	0 ~ 255	0 ~ 255	Ref.1
1	<b>Network ID</b>	0 ~ 255	0 ~ 255	Ref.2
2	<b>Command / Security</b>	0 ~ 255	0 ~ 255	Ref.3
3	<b>Left Wheel</b>	-100 ~ +100 (+fwd, -bwd)	-100 ~ 100 %	
4	<b>Right Wheel</b>			
5	<b>Left LED</b>	0 (off) ~ 7	0 (off) ~ 7	Ref.4
6	<b>Right LED</b>			
7	<b>Buzzer High</b>	0(off) 1 ~ 16777215	0(off) 1.00 Hz ~ 167.77215 KHz,	Ref.5
8	<b>Buzzer Middle</b>			
9	<b>Buzzer Low</b>			
10	<b>Musical Note</b>	0(off) ~ 88 (piano key)	0 (off) ~ 88	Ref.6
11	<b>Line Tracer Mode/Speed</b>	0x11 ~ 0x6A 0x0?(off)	0x11 ~ 0x6A 0x0?(off)	Ref.7
12	<b>Proximity IR Current</b>	0 ~ 7 (default 2)	0 ~ 7 (default 2)	Ref.8
13	<b>G-Range, Bandwidth</b>	0 ~ 3 (default 0), 0 ~ 8 (default 3)	0 ~ 3 (default 0), 0 ~ 8 (default 3)	Ref.9
14	<b>IO Mode(A, B)</b>	0 ~ 127	0 ~ 127	Ref.10
15	<b>Output A</b>	0 ~ 255	0 ~ 255	
16	<b>Output B</b>			
17	<b>Wheel Balance</b>	-128 ~ 127	-128 ~ 127	Ref.11
18	<b>Input Pull</b>	0~16		Ref.12
19	<b>not assigned</b>			

#### Ref.1) Version, Topology

bit	7	6	5	4	3	2	1	0
value	<b>Version Number</b> Default value: 0x0				<b>Topology Type</b> Default value: 0x0			

Indicating the networking type and create the commanded network topology type in the device (Hamster). Currently, only the default is supported.

Each topology type needs corresponding firmware (with different version number). The packet definition can be changed depending on the version number.

(Caution) Make sure to confirm from the Sensors packet that the device (Hamster) can create the commanded network topology type before proceed.

#### Topology type

0	None (connect only to hub)
1	Daisy Chain
2	Star
3	Extended Star

### Ref.2) Network ID

Regardless of Topology type, if ID == 0x00, the packet is for yourself.

#### Topology 0: None

ID is always 0x00.

#### Topology 1: Daisy Chain

bit	7	6	5	4	3	2	1	0
value	Node 0 ~ 255							

```
if (packet[1] == 0x00)
    the packet is for yourself
else
    packet[1] --; // deliver to children
```

#### Topology 2: Star

bit	7	6	5	4	3	2	1	0
value	Node 0 ~ 255							

```
if (packet[1] == 0x00)
    the packet is for yourself
else {
    child = packet[1] - 1; // index of child to which the packet should be delivered
    packet[1] = 0x00;
}
```

#### Topology 3: Extended Star

bit	7	6	5	4	3	2	1	0
value	Depth 4		Depth 3		Depth 2		Depth 1	

Max level of depth == 4. Max number of children == 3.  
Each Depth, first child == 1, second child == 2, third child == 3.

```
if (packet[1] == 0x00)
    the packet is for yourself
else {
    child = (packet[1] & 0x03) - 1; // index of child to which the packet should be delivered
    packet[1] >>= 2;
}
```

### Ref.3) Command, Security

bit	7	6	5	4	3	2	1	0
value	Command				Security			

If Security == 0, not encrypted.

If Security != 0, encrypted. (Different encryption methods depending on Security value)

```
int command = (packet[2] >> 4) & 0x0f;
```

```
int security = packet[2] & 0x0f;
```

### Command Types

1. Motoring packet: 0x1?

If (packet[2] >> 4) & 0x0f == 0x01, it is Effectors packet.

2. Bluetooth reset

byte	detail	value to robot	user input
2	Command+Security	0xF?	0xF?

If (packet[2] >> 4) & 0x0f == 0x0f, Reset the Bluetooth

3. Permanent storage of Wheel Balance value

byte	detail	value to robot	user input
2	Command+Security	0xE?	0xE?
17	Wheel Balance	-128 ~ 127	-128 ~ 127

If (packet[2] >> 4) & 0x0f == 0x0e, current Wheel Balance value (packet[17]) is stored permanently and applied immediately affecting all future wheel speeds.

The positive Wheel Balance value will make wheels move bias to the right, and the negative value will make wheels move bias to the left. Therefore, if Hamster moves left with an equal speed value to both wheels, a positive Wheel Balance value will make the motion straighter.

(Caution) The Wheel Balance value is added to the permanently stored Wheel Balance value before applied to the motors. So, if you want to change Wheel Balance value, you need to aware of the permanently stored Wheel Balance value (usually setting to 0 before changing Wheel Balance value).

4. Effectors command timeout (life-time)

byte	detail	value to robot	user input
2	Command+Security	0xD?	0xD?
3	Life-time Constant	0(infinite) or 1~255	25 msec*255 또는 무한

Effectors packet has a time limit. After the time limit, Effector packet is reset to 0 automatically causing Hamster to stop motors and turn off LEDs and sounds. Therefore, next Effectors packet has to be transmitted before the reset if you want to control the robot.

### Ref.4) Left/Right LED

bit	7	6	5	4	3	2	1	0
value	white	yellow	magenta	red	cyan	green	blue	off

### Ref.5) Buzzer

Formula) Frequency = Buzzer / 100 (accuracy: 1/100)

Buzzer value (24 bit): 1 ~ 16777215

ex) 100 == 1.00 hz, 4800000 == 48,000.00 hz, 16777215 == 167,772.15 hz

(Caution) If value == 0, Buzzer is off.

### Ref.6) Musical Note

Musical notes are produced by Buzzer (accuracy < 0.01%).

Activated only when Buzzer value == 0(off).

Values (1~88) are keys of Piano using the 12-tone Equal Temperament.

Frequency range: A0(27.50Hz) ~ C8(4186.01Hz). Refer to Appendix#2.

ex) 54 → 5th Octave D (D5: 587.330 hz)

ex) 70 → 6th Octave F# (F#6: 1479.98 hz)

ex) 40 → 4th Octave C (middle C: C4: 261.626 hz)

### Ref.7) Line Tracer Mode/Speed

\* Hamster has line tracing algorithms in the firmware in order to support high speed line tracing without the inherent network (Bluetooth communication) delay. These algorithms use feedback control with sampling rate > 50 hz.

\* The optimal line width (black/white) is 9.0 mm.

\* In case of 90 degree turn, make sure to cut the corner to 45 degree such that the thickest part of line is less than 10 mm.

\* The between left and right sensors is 16mm.

bit	7	6	5	4	3	2	1	0
value	command	mode				speed		

**mode: 0** Inactive (**Off mode**)

**mode: 1** left sensor/black (**Black\_Left mode**)

**mode: 2** right sensor/white (**Black\_Right mode**)

**mode: 3** both sensors/black (**Black\_Both mode**)

**mode: 4** one segment left turn/black (**Black\_Segment\_Left mode**)

**mode: 5** one segment right turn/black (**Black\_Segment\_Right mode**)

**mode: 6** one segment forward/black (**Black\_Segment\_Forward mode**)

**mode: 7** one segment U-turn/black (**Black\_Segment\_UTurn mode**)

**mode: 8** left sensor/white (**White\_Left mode**)

**mode: 9** right sensor/white (**White\_Right mode**)

**mode: 10** both sensors/white (**White\_Both mode**)

**mode: 11** one segment left turn/white (**White\_Segment\_Left mode**)

**mode: 12** one segment right turn/white (**White\_Segment\_Right mode**)

**mode: 13** one segment forward/white (**White\_Segment\_Forward mode**)

**mode: 14** one segment U-turn/white (**White\_Segment\_UTurn mode**)

\* mode and speed are independent of each other.

\* When sending a new mode, command (bit 7) should be toggled between 1 and 0.

\* Speed has 8 levels (0 ~ 7). 7 is max speed(100%) and 0 is the min speed.

```
check = (packet[11] >> 3) & 0x1f;
```

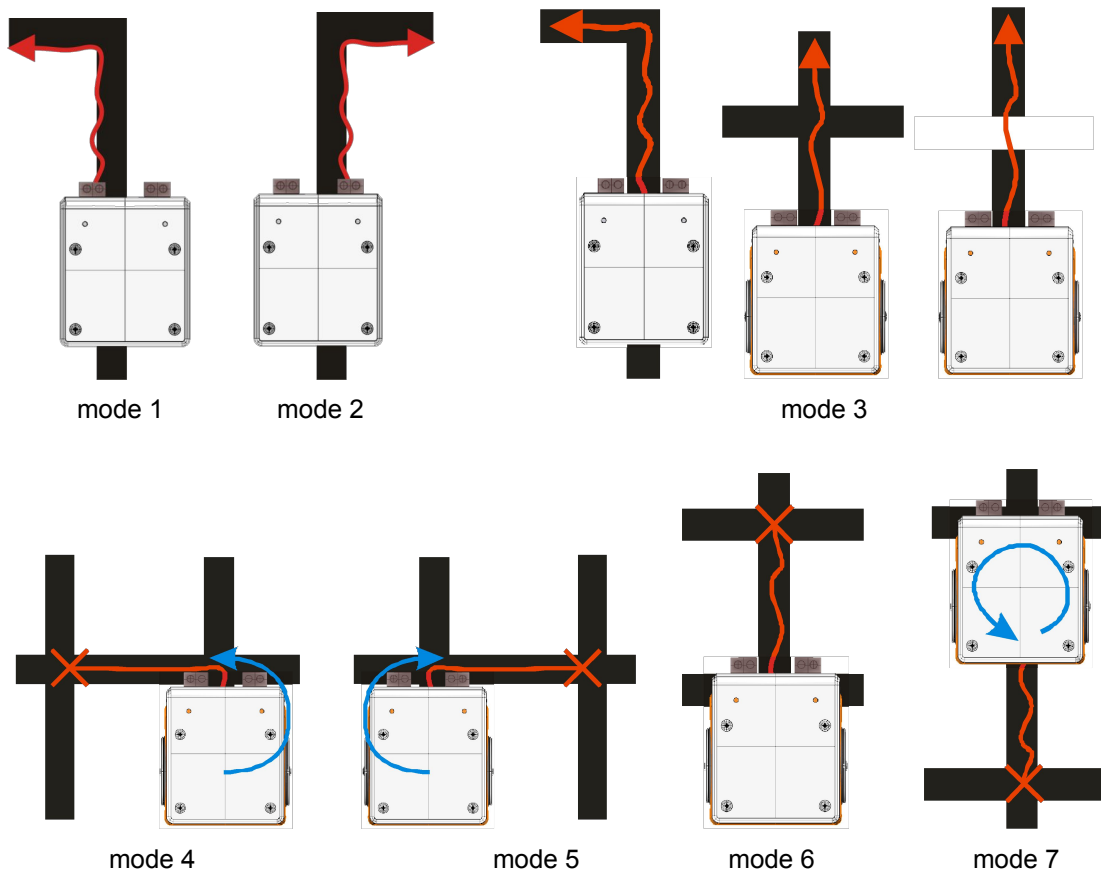
```
mode = check & 0x0f;
speed = packet[11] & 0x07;
```

If “check” is different from the “previous check”, there is a new command

- \* If mode != 0, Left/Right Wheel commands are ignored.
- \* If mode == 0, line tracing speed is ignored and Hamster moves by Left/Right Wheel commands.

The following shows the motions of each mode (Hamster stops at X for modes 4~7).

\* Note that modes 8~14 is the same as modes 1~7 except the line/floor colors(black/white) are flipped to white/black.



### Ref.8) Proximity IR Current

Current of IR Sensors (default: 0x0 or 0x02 = 10mA)

- If needed to detect farther, change it to higher current.
- Default setting (0x02) is optimal for detecting distance < 10cm.

0x01	0x0 or 0x02	0x03	0x04	0x05	0x06	0x07
5 mA	10 mA	20 mA	50 mA	100 mA	150 mA	200 mA

### Ref.9) G-Range / Band Width

bit	7	6	5	4	3	2	1	0
value	G-Range				Bandwidth			

Accelerometer: Gravity measuring range (default: 0x0 = 2g where g is 9.81 m/sec<sup>2</sup>)

- Default value (2g) is to detect the orientation of Hamster.
- If needed to measure impact or fast motion, use higher measuring range.
- If you double the measuring range, the measuring value is halved.

0x00	0x10	0x20	0x30
2g	4g	8g	16g

Accelerometer: Bandwidth (default: 0x3 = 31.25 hz)

- setting the responsiveness of accelerometer.
- Higher bandwidth → faster response but more noise

0x01	0x02	0x00 or 0x03	0x04	0x05	0x06	0x07	0x08
7.81 hz	15.63 hz	31.25 hz	62.5 hz	125 hz	250 hz	500 hz	1000 hz

### Ref.10) IO Mode

External Input/Output mode(currently supports only 0x00~0x05 and 0x08, 0x09, 0x0A modes)

bit	7	6	5	4	3	2	1	0
value	Port A				Port B			
default	ADC mode, 0x0				ADC mode, 0x0			

Port A and Port B are independent of each other.

**0x00 ADC** (Analog-to-Digital)

**0x01 DI** (Digital Input)

**0x08 SERVO** (Analog Servo Control)

**0x09 PWM** (Digital-to-Analog)

**0x0A DO** (Digital Output)

**ADC** (Analog-to-Digital) Mode: 0x00

Measures input voltage with 8-bit ADC.

Max input voltage is ~3.7volt → 255(0xFF)

**DI** (Digital Input) Mode: 0x01

Detect input voltage to either 0 or 1.

1 if input voltage > 3.7/2 (~1.8 volt)

0 otherwise

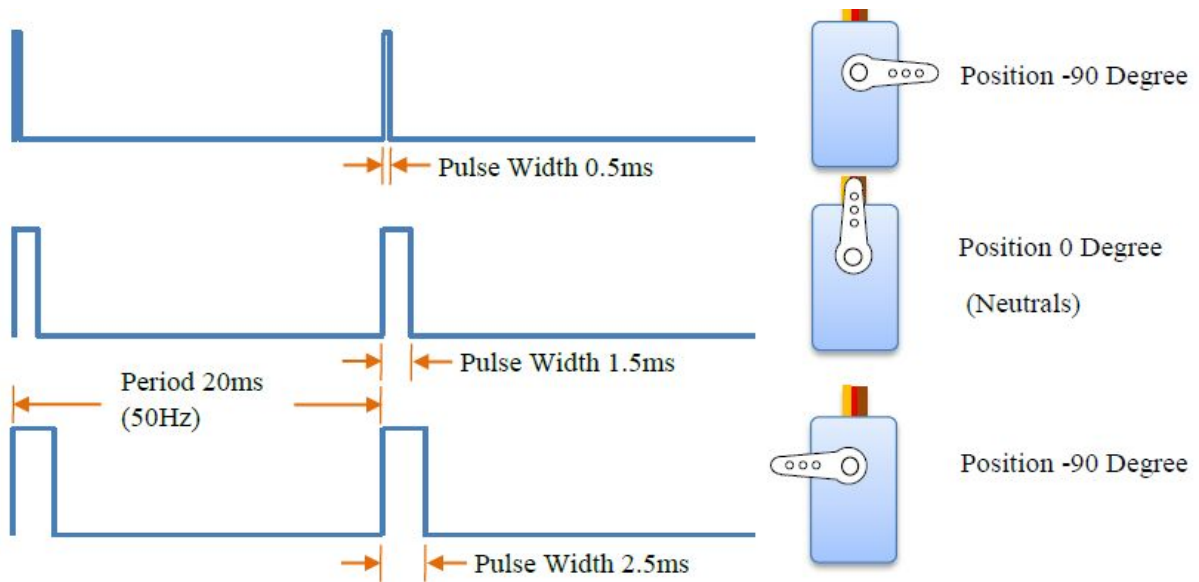
**SERVO** (Analog servo) Output Mode: 0x08

Generating PWM signal(mode = 8) for external Servo control

Port A	1 ~ 180	0(off), 90(center)	1deg=1.0ms, 90deg=1.5ms, 180deg=2.0ms
Port B	1 ~ 180	0(off), 90(center)	1deg=1.0ms, 90deg=1.5ms, 180deg=2.0ms

\* If value == 0(off) → no pulse

\* If value > 180, pulse width limits to 2.5 ms



**PWM** (Digital-to-Analog) Output Mode: 0x09

Output: PWM signal's Duty value

If value > 100(0x64), output is 1 and PWM pulse period is 20 msec.

Therefore, if Duty value is 50%(50, 0x32), output is 0 for 10 msec, then output is 1 for the next 10msec.

**DO** (Digital Output) Mode: 0x0A

If value is not 0, output is 'high'.

### Ref.11) Wheel Balance

Correction of the ratio of wheel(motor) speeds to achieve straight forward motion when same wheel speeds are given to both wheels.

Default value is 0 and any value is reset to when Hamster is off.

(Caution) If you want to store Wheel Balance value permanently, you have set Command(Byte 2) value to 0xE0.

### Ref.12) Port Pull Up/Down (incomplete)

Port A/B extension setting: changing Pull-up or Pull-down of internal resistance(~10 kohm).

Default is Pull-up.

bit	7	6	5	4	3	2	1	0
value	reserved				Port B		Port A	

#### Port A

0: Pull-up, connected to VCC

1: Pull-down, connected to GND

2, 3: No-pull, no connection

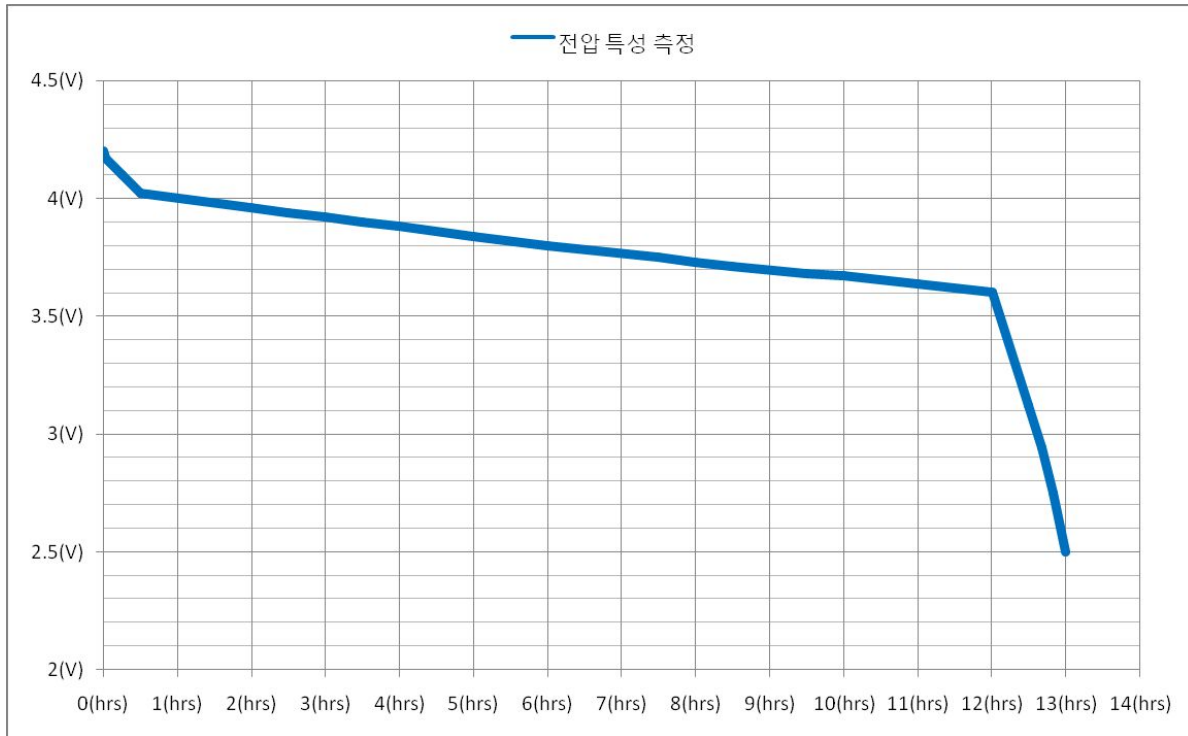
#### Port B

0: Pull-up, connected to VCC

1: Pull-down, connected to GND

2, 3: No-pull, no connection

## Appendix 1: Hamster Battery Usage Graph



\* According to this graph, the max standby time is 12 hours

\* Note: Operational time is approximately 1 hour.

## Appendix 2: Hamster Musical Notes and Piano Keys with Frequency

Key number	Notation	Frequency(Hz)Hz
88	C8 - Last tone	4186.01
87	B7	3951.07
	A $\sharp$ 7/B $\flat$ 7	3729.31
85	A7	3520.00
	G $\sharp$ 7/A $\flat$ 7	3322.44
83	G7	3135.96
	F $\sharp$ 7/G $\flat$ 7	2959.96
81	F7	2793.83
80	E7	2637.02
	D $\sharp$ 7/E $\flat$ 7	2489.02
78	D7	2349.32
	C $\sharp$ 7/D $\flat$ 7	2217.46
76	C7	2093.00
75	B6	1975.53
	A $\sharp$ 6/B $\flat$ 6	1864.66



73	A6	1760.00
	G $\sharp$ 6/A $\flat$ 6	1661.22
71	G6	1567.98
	F $\sharp$ 6/G $\flat$ 6	1479.98
69	F6	1396.91
68	E6	1318.51
	D $\sharp$ 6/E $\flat$ 6	1244.51
66	D6	1174.66
	C $\sharp$ 6/D $\flat$ 6	1108.73
64	C6 (high C)	1046.50
63	B5	987.767
	A $\sharp$ 5/B $\flat$ 5	932.328
61	A5	880.000
	G $\sharp$ 5/A $\flat$ 5	830.609
59	G5	783.991
	F $\sharp$ 5/G $\flat$ 5	739.989
57	F5	698.456
56	E5	659.255
	D $\sharp$ 5/E $\flat$ 5	622.254
54	D5	587.330
	C $\sharp$ 5/D $\flat$ 5	554.365
52	C5	523.251
51	B4	493.883
	A $\sharp$ 4/B $\flat$ 4	466.164
49	<b>A4 concert pitch</b>	440.000
	G $\sharp$ 4/A $\flat$ 4	415.305
47	G4	391.995
	F $\sharp$ 4/G $\flat$ 4	369.994
45	F4	349.228
44	E4	329.628
	D $\sharp$ 4/E $\flat$ 4	311.127
42	D4	293.665
	C $\sharp$ 4/D $\flat$ 4	277.183
40	C4 (middle C)	261.626
39	B3	246.942
	A $\sharp$ 3/B $\flat$ 3	233.082
37	A3	220.000
	G $\sharp$ 3/A $\flat$ 3	207.652
35	G3	195.998
	F $\sharp$ 3/G $\flat$ 3	184.997
33	F3	174.614
32	E3	164.814
	D $\sharp$ 3/E $\flat$ 3	155.563
30	D3	146.832
	C $\sharp$ 3/D $\flat$ 3	138.591

28	C3	130.813
27	B2	123.471
	A $\sharp$ 2/B $\flat$ 2	116.541
25	A2	110.000
	G $\sharp$ 2/A $\flat$ 2	103.826
23	G2	97.9989
	F $\sharp$ 2/G $\flat$ 2	92.4986
21	F2	87.3071
20	E2	82.4069
	D $\sharp$ 2/E $\flat$ 2	77.7817
18	D2	73.4162
	C $\sharp$ 2/D $\flat$ 2	69.2957
16	C2 (low C)	65.4064
15	B1	61.7354
	A $\sharp$ 1/B $\flat$ 1	58.2705
13	A1	55.0000
	G $\sharp$ 1/A $\flat$ 1	51.9130
11	G1	48.9995
	F $\sharp$ 1/G $\flat$ 1	46.2493
9	F1	43.6536
8	E1	41.2035
	D $\sharp$ 1/E $\flat$ 1	38.8909
6	D1	36.7081
	C $\sharp$ 1/D $\flat$ 1	34.6479
4	C1	32.7032
3	B0	30.8677
	A $\sharp$ 0/B $\flat$ 0	29.1353
1	A0 - First tone	27.5000