

# Honeywell Evotouch Protocol

The unbreakable broken

## 1. Physical Layer

If you are using other than CC1101, please refer to CC1101 datasheet.

### 1.1. CC1101 settings

Registers in *italics* are not programmed, CC1101 defaults used.

Register	Value	Function
00	0D	RF channel*
01	2E	
02	2E	
<i>03</i>	07	
<i>04</i>	D3	
<i>05</i>	91	
<i>06</i>	FF	
07	00	
08	32	
<i>09</i>	00	
<i>0A</i>	00	
0B	06	
0C	00	
0D	21	
0E	65	
0F	CC	
10	6A	
11	83	
12	10	

Register	Value	Function
13	22	RF power**
14	F8	
15	50	
<i>16</i>	07	
<i>17</i>	30	
18	18	
19	16	
1A	6C	
1B	43	
1C	40	
1D	91	
23	E9	
24	2A	
25	00	
26	1F	
29	59	
2C	81	
2D	35	
2E	09	
3E	C0	

\*The default value for *RF channel* is 0xCC, but it is factory programmable.

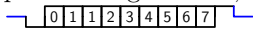
\*\*The default value for *RF power* is 0xC0, but it is factory programmable.

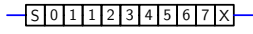
## 2. Channel Coding

The channel coding for transmission and reception is absolutely the same.

### 2.1. Basics

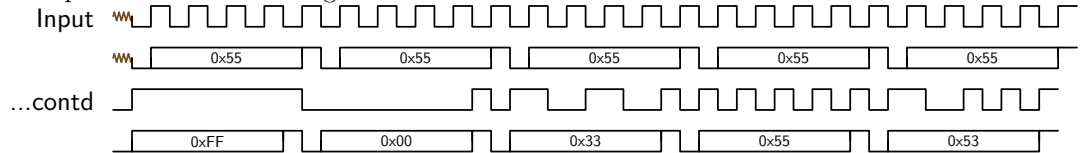
Serial port settings: 38400, 8 bits LSB first, no parity, 1 start bit, 1 stop bit

Input 

Input 

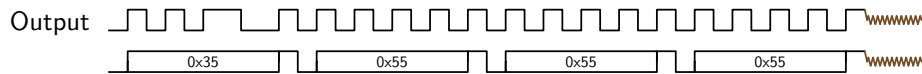
### 2.2. Preamble

The preamble of the message as follows



For practical reasons I suggest to sync on 00 33 55 53. Values 00, 33 and 53 are Manchester-breaking, anyway.

### 2.3. End of block



For practical reasons I suggest to sync on Manchester-breaking word 35.

### 2.4. Manchester-decoding

To decode the actual message a simple Manchester decoding procedure should be used. If Manchester-decoding fails, the message should not be processed.

Input	Decoded
10	0
01	1

For practical reasons I suggest to use the following table, high nibble first, low nibble last:

55	F	85	7
56	E	86	6
59	D	89	5
5A	C	8A	4
65	B	95	3
66	A	96	2
69	9	99	1
6A	8	9A	0

Example: Received 55 95 decoded to F3.

### 3. Message format

From this point we are speaking of the Manchester-decoded message block.

#### 3.1. Checksum

Checksum is the last byte of the message block.

$$\sum_{i=0}^{len} rcv_i = 0_b$$

If checksum verification fails, the message should not be processed.

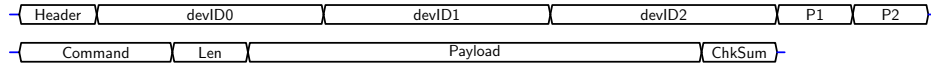
#### 3.2. Byte order, data formats

For each multi-byte (16 bit unsigned, 16 bit signed, 24 bit) values the byte order is little-endian (MSB first).

The data format for temperatures is a 16 bit signed integer value, the units are  $\frac{1}{100}$  Celsius; if not otherwise indicated.

#### 3.3. Message Header

The message block starts with a header byte, that contains additional flags and describes which optional content is included in the message header.




Header byte	8	required	
Device id 0	24	optional	Flags* bit 0
Device id 1	24	optional	Flags* bit 1
Device id 2	24	optional	Flags* bit 2
Param 1	8	optional	Header bit 1
Param 2	8	optional	Header bit 0
Command	16	required	Command code
Payload length	8	required	Payload length
Payload	var	required	Payload length $\times$ 8
Checksum	1	required	Checksum to 0x00

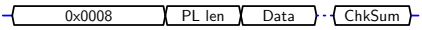
\*Flags can be retrieved from Header[5:2] as follows:

0	0F	8	47
1	0C	9	44
2	0D	A	45
3	0B	B	43
4	27	C	17
5	24	D	14
6	25	E	15
7	23	F	13

## Command 0x0001

Output 

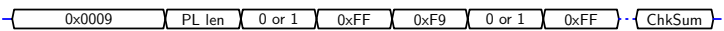
## 0x0008: Hotwater or Boiler Relay

Output 

**Data:**

2 × something

## 0x0009: Hotwater or Boiler Relay

Output 

**Data:**

something

**Data:**

something

**Data:**

something

**Data:**

something

**Data:**

something

## 0x000A: Set HR80 local parameters

Output 

### Zone:

Zone to apply for

### Flags:

- 0: Local override enabled
- 1: Window function enabled
- 2: Zone something enabled

### minTemp:

Minimum setting temperature

### maxTemp:

Maximum setting temperature

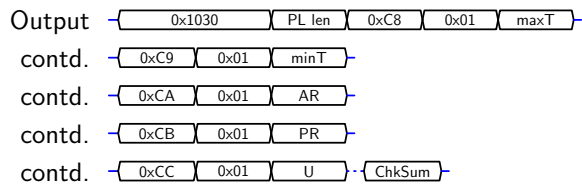
## 0x0016: DUNNO

Output 

### Data:

something

## 0x1030: DUNNO



**Fixed:**

0xC8

**Fixed:**

0x01

**maxTemp (maxT):**

Maximum Temperature / 100

**Fixed:**

0xC8

**Fixed:**

0x01

**minTemp (minT):**

Minimum Temperature / 100

**Fixed:**

0xC8

**Fixed:**

0x01

**accRun (AR):**

Actuator running time

**Fixed:**

0xC8

**Fixed:**

0x01

**pumpRun (PR):**

Pump running time

**Fixed:**

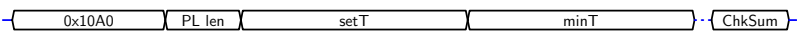
0xC8

**Fixed:**

0x01

**dunno (U):**

## 0x10A0: Hotwater temperature setting

Output 

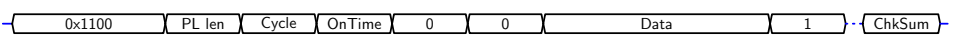
### setTemp (setT):

Hotwater setting temperature

### minTemp (minT):

Hotwater minimum temperature

## 0x1100: Hotwater or Boiler Relay

Output 

### Cycle Rate (Cycle):

$4 \times \text{Cycle Rate}$

### Minimum On-Time (OnTime):

$4 \times \text{Minimum On-Time}$

### Data:

something

### Data:

something


### Data:

something


### Data:

something

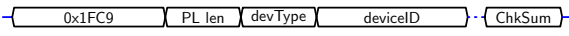
## 0x1260: DUNNO

Output 

## 0x1FC9: Binding

Output 

## 0x1FC9: Binding

Output 

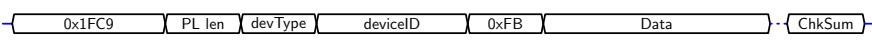
### devType:

See device types 0x2309 = evoTouch

### deviceID:

Source Device ID

## 0x1FC9: Binding

Output 

### devType:

See device types 0x2309 = evoTouch

### deviceID:

Source Device ID

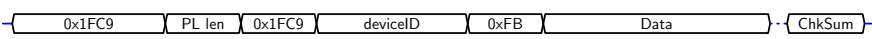
### Data:

0xFB

### Data:

Device type 0x3150

## 0x1FC9: Binding

Output 

### devType:

See device types

### deviceID:

Source Device ID

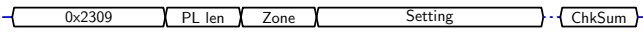
### Data:

0xFB

### Data:

Device type 0x3150

### 0x2309: Zone Setpoint Setting

Output 

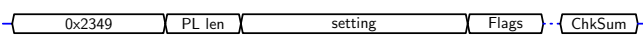
#### Zone:

Zone

#### Setting:

Setpoint temperature in  $\frac{1}{100}$  Celsius

### 0x2349: Zone Temp Setting

Output 

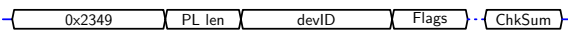
#### setting:

Setting / 100

#### Flags:

0 or 3

### 0x2349: Incoming 3

Output 

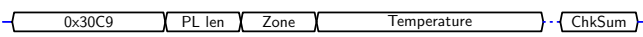
#### devID:

Setting / 100

#### Flags:

0 or 3

### 0x30C9: Zone temperature distribution

Output 

#### Zone:

Zone

#### Temperature:

Temperature in C \* 100

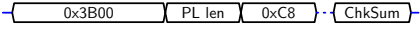
### 0x3150: Heat demand timing related

Output 

#### Data:

2 × something

### Command 0x3B00

Output 

#### Data:

Always 0xC8