Wireless temperature sensor decoding using Arduino

For reading:

Code for Arduino how to decode pulses:
http://arduino.cc/forum/index.php?topic=67545.30
http://arduino.cc/forum/index.php/topic,68773.0.html

CRC
http://www.mikrocontroller.net/topic/250207
http://www.mikrocontroller.net/topic/38129

FUNK- WETTERSTATION FUNK-TEMPERATUR
http://tfa-dostmann.de/index.php?id=57
FUNKTHERMOMETER TFA 30.3034.01
Funk-Temperatursender TFA 30.3120.90 Kat. Nr. 30.3163

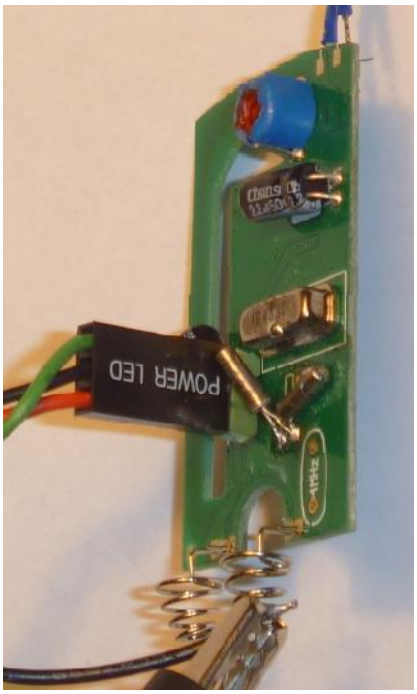
Sends 1 value/ min.
Intelligent – changes address every time battery is inserted.

Transmitter hermetically sealed. Very nice.
2AA batteries. Power consumption: 35 mA TX. 2...10uA sleep

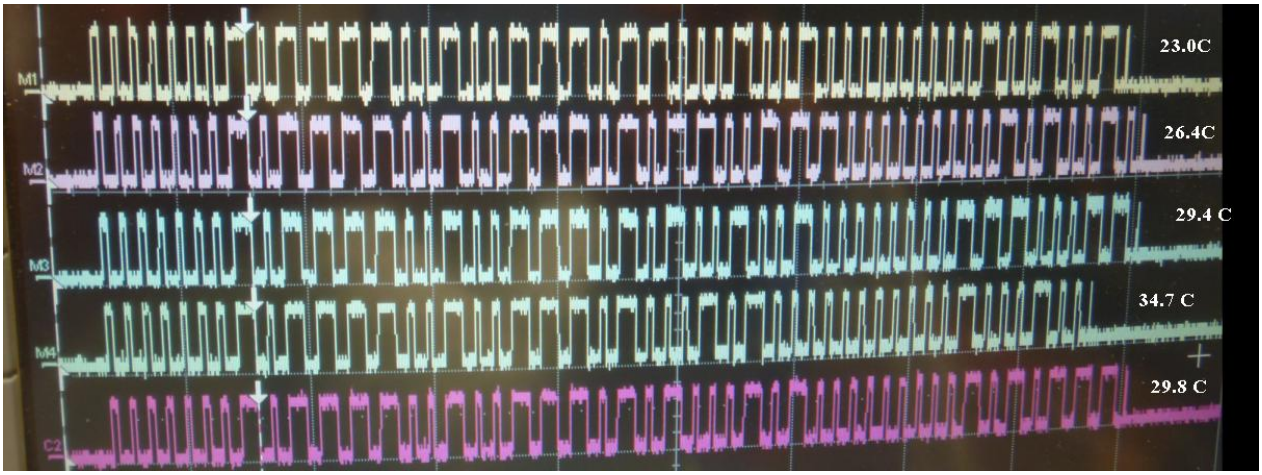
20 m range checked OK.

432 MHz Temperature Sensor Transmitter

Thermistor on long wires to put to cold/hot places. Can be replaced by a variable resistor
On delivery the red LED was on all the time. 32kHz quartz was broken. Had to resolder.
There is a 3 pin header. Two pins are +3V and GND, third is signal sent to transmitter.



Oscilloscope traces:



short pulse length 460 us
long length 1435 us ratio 3
pause 987 us
47 bits sent

433 MHz temperature sensor emulation with Arduino

Does not work perfectly. Works only ca 1 min while receiver is searching sensors.
This was because receiver only listens in a short time interval every minute and turns off receiver power to save battery.

Arduino pin 13 connected to data pin via 20 Ohm resistor as transmitter has 3 V levels.

32kHz quartz gives frequency resolution 30 us. So take > +- 100 us window for detection.

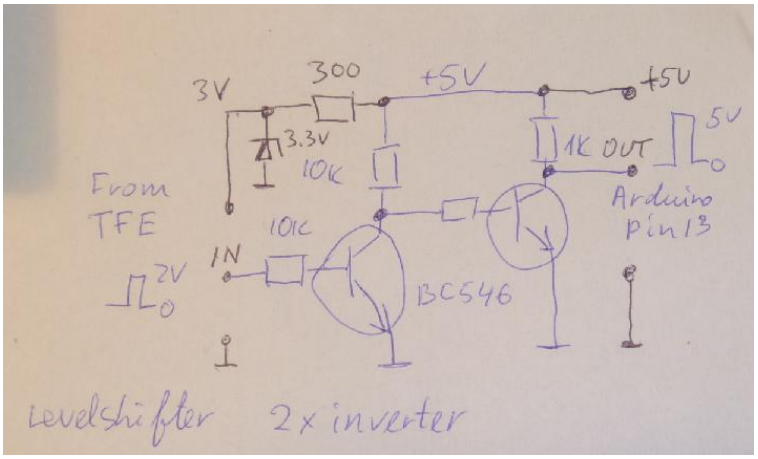
```
int tpin=13;  // 433 MHz RF transmitter hooked to this pin
String report="";
void setup() {
  Serial.begin(9600); Serial.flush();
  Serial.println ("Make sure there is no echo from USB host back to Arduino, buffer will overload
and commanding will not work");
}
  void I() {     // sends one
           digitalWrite(tpin, HIGH); delayMicroseconds(1435);   digitalWrite(tpin, LOW);
delayMicroseconds(987); }
  void O() {    // sends zero
           digitalWrite(tpin, HIGH); delayMicroseconds(460);   digitalWrite(tpin, LOW);
delayMicroseconds(987); }
  void S1() {
O();O();O();O();O();O();O();I();O();I();I();I();I();O();O();I();O();O();O();I();I();I();O();I();} //
sensor ID 1

void loop() {
pinMode(tpin, OUTPUT);
S1();
O();I();I();I();I();I(); O();I();O();O();O();O();O();O();O();O();O();I();I();O();I();O();O();I();O();
// 24.2
digitalWrite(tpin, HIGH); delayMicroseconds(20);   digitalWrite(tpin, LOW);
pinMode(tpin, INPUT);
Serial.println("24.2");
delay(10000);
}
```

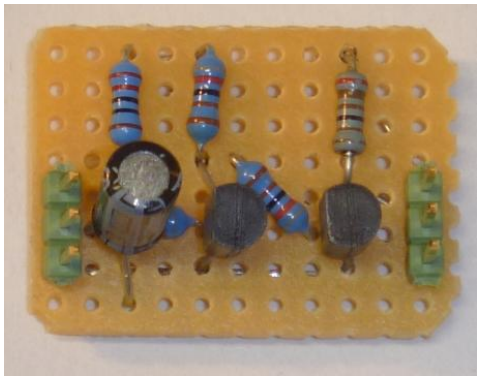
Level convertor from 3 V to TTL

Next step is to connect Arduino to the TFE 432 MHz receiver.

Some other 432 MHz receiver boards were tested, but TFE was the best in sensitivity range.

TFE receiver has low-power OP LM358N. Pin 1 is output. 2 V level is not enough to drive 5V Arduino. Need level converter. Use 2 transistors because one would invert signal.





To save battery TFE display unit turns off power to built in receiver and switches it on every minute for short time ca 0.3 s. It knows when to listen from first sensor transmission..

This needs to be blocked. Receiver must be on all time.
This is done by supplying 3 V to receiver power line.

It is necessary to leave AA batteries in receiver or reception gets spikes, probably from PC.

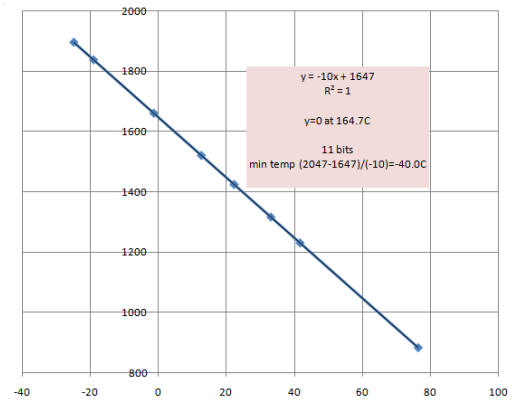
Arduino code to read out what transmitter sends

```
byte pin = 13;
byte whole_array[116];
byte value;
byte array_pos=0;
unsigned long duration;
char numStr[8];

void setup()
{
    pinMode(pin, INPUT);
   Serial.begin(9600); Serial.flush();
}

void loop()
{
  duration = pulseIn(pin, HIGH);
//      Serial.println(duration);
  if ((duration>300)&&(duration<600)) {
    whole_array[array_pos] = 0;
    array_pos++;

        Serial.print("0");
  }
  if ((duration>1200)&&(duration<1500)){
    whole_array[array_pos] = 1;
    array_pos++;
        Serial.print("1");
  }
 if (array_pos==11) { Serial.print(" "); }
 if (array_pos==19) { Serial.print(" "); }
 if (array_pos==20) { Serial.print(" "); }
 if (array_pos==31) { Serial.print(" "); }
 if (array_pos==39) { Serial.print(" "); }

 if (array_pos==47) { Serial.println(); array_pos=0; }
//delay (100);
}
```

Truth table recorded from Temperature sensor

```
over    00000001011 11111101 1 00000000000 00000000 10101000
76.3    00000001011 11111101 1 01101110100 00000000 11011010  884
70      00000001011 11111101 1 01110110011 00000000 11011110
68.8    00000001011 11111101 1 01110111111 00000000 01101010
67      00000001011 11111101 1 01111010001 00000000 01010010
42.3    00000001011 11111101 1 10011001000 00000000 00011100
41.6    00000001011 11111101 1 10011001111 00000000 10110010  1231
39.7    00000001011 11111101 1 10011100010 00000000 00101110
38.7    00000001011 11111101 1 10011101100 00000000 01000011
34.3    00000001011 11111101 1 10100011000 00000000 10011110
33.6    00000001011 11111101 1 10100011111 00000000 00110000
33.5    00000001011 11111101 1 10100100000 00000000 00011011
33      00000001011 11111101 1 10100100101 00000000 01101100  1317
32.3    00000001011 11111101 1 10100101100 00000000 10101111
26.3    00000001011 11111101 1 10101101000 00000000 10100101
23.6    00000001011 11111101 1 10110000011 00000000 11001001
22.9    00000001011 11111101 1 10110001010 00000000 00001010
22.6    00000001011 11111101 1 10110001101 00000000 10100100
22.4    00000001011 11111101 1 10110001111 00000000 01111101
22.2    00000001011 11111101 1 10110010001 00000000 01111110  1425
21.9    00000001011 11111101 1 10110010100 00000000 00001001
21.8    00000001011 11111101 1 10110010101 00000000 11111101
21.4    00000001011 11111101 1 10110011001 00000000 01001001
20.2    00000001011 11111101 1 10110100101 00000000 01001111
19.2    00000001011 11111101 1 10110101111 00000000 10100001
15.5    00000001011 11111101 1 10111010111 00000000 10101101
12.6    00000001011 11111101 1 10111110001 00000000 00101011  1521
10.9    00000001011 11111101 1 11000000010 00000000 11010100
9.6     00000001011 11111101 1 11000001111 00000000 10010100
5.8     00000001011 11111101 1 11000110101 00000000 11001000
2       00000001011 11111101 1 11001011011 00000000 11110000
-1.4    00000001011 11111101 1 11001111101 00000000 01110110  1661
-5.4    00000001011 11111101 1 11010100101 00000000 10000101
-9      00000001011 11111101 1 11011001001 00000000 01100100
-13     00000001011 11111101 1 11011110001 00000000 11100001
-19.1   00000001011 11111101 1 11100101110 00000000 11111010  1838
-19.7   00000001011 11111101 1 11100110100 00000000 01111010
-24.9   00000001011 11111101 1 11101101000 00000000 00101001  1896
over    00000001011 11111101 1 00000000000 00000000 10101000
```

Took half a night to figure out that this box actually sends ADC value. It is already linearised.



11 bits model
8 bits random id changes on power up
1 bit battery status 1=OK 0=low
11 bits temperature
8 bits humidity?
8 bits ? parity checksum? CRC? Further study needed.

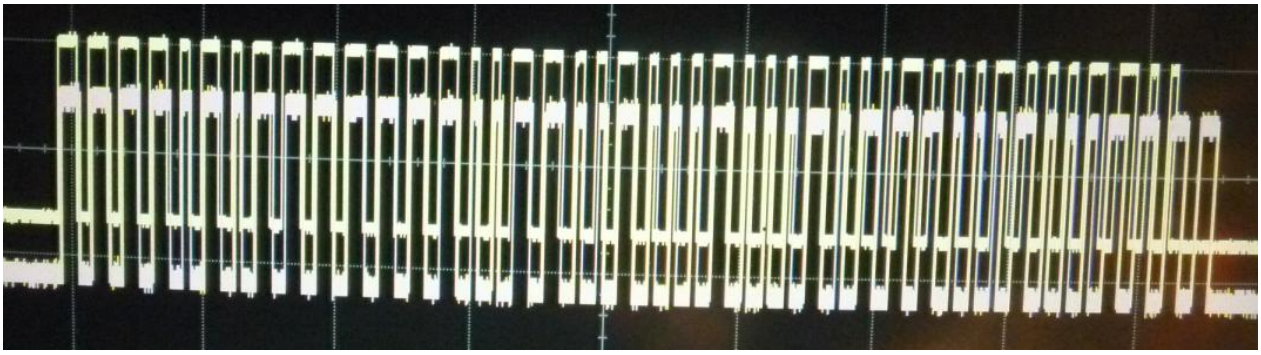
Arduino code that calculates temperature from sensor data

```
// Funksensor TFA, 433 MHz Temperatur-Außensender 30.3163, receiver 30.3034.01
// 00000001011 11111101 1 10110001111 00000000 01111101       22.4C
// 00000001011 11111101 1 10110010001 00000000 01111110  1425 22.2C

// 11 bits  model
// 8  bits  random id changes on power up
// 1  bit battery status 1=OK 0=low
// 11 bits  temperature
// 8  bits  humidity?
// 8  bits  ? parity checksum? CRC?

byte pin = 13; // from receiver. Between receiver and Arduino is 2 transistor 3V to TTL voltage
converter
byte ar[116]; byte pos=0; unsigned long dur; //array, position in array, pulse duration

void setup() { pinMode(pin, INPUT);  Serial.begin(9600); Serial.flush(); }

void loop() { b: pos=0;
  dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<600)) {ar[pos] = 0; pos++;
   dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<600)) {ar[pos] = 0; pos++;
    dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<600)) {ar[pos] = 0; pos++;
     dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<600)) {ar[pos] = 0; pos++;
      dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<600)) {ar[pos] = 0; pos++;
       dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<600)) {ar[pos] = 0; pos++;
        dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<600)) {ar[pos] = 0; pos++;
         dur = pulseIn(pin, HIGH); if ((dur>1200)&&(dur<1500)) {ar[pos] = 1; pos++;
          dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<600)) {ar[pos] = 0; pos++;
           dur = pulseIn(pin, HIGH); if ((dur>1200)&&(dur<1500)) {ar[pos] = 1; pos++;
            dur = pulseIn(pin, HIGH); if ((dur>1200)&&(dur<1500)) {ar[pos] = 1; pos++;
            }else goto b;
           }else goto b;
          }else goto b;
         }else goto b;
        }else goto b;
       }else goto b;
      }else goto b;
     }else goto b;
    }else goto b;
   }else goto b;
  }else goto b;
 //  Serial.print("data "); // if one gets to this point then message is most probably correct
 for (int i=1; i <= 36; i++){
  dur = pulseIn(pin, HIGH);
  if ((dur>300)&&(dur<600)){ar[pos] = 0; pos++; Serial.print("0");}
  if ((dur>1200)&&(dur<1500)){ar[pos] = 1; pos++; Serial.print("1");}

 if (pos==11) { Serial.print(" "); }
 if (pos==19) { Serial.print(" "); }
 if (pos==20) { Serial.print(" "); }
 if (pos==31) { Serial.print(" "); }
 if (pos==39) { Serial.print(" "); }
 if (pos==47) { Serial.print(" "); }
 }
 int dev=0; for (int i=0; i <= 7; i++) {dev=dev << 1; dev=dev+ar[11+i]; }   // Device ID
 // if (whole_array[20+i]==1) Serial.print("1"); if (whole_array[20+i]==0) Serial.print("0");
 Serial.print(dev);

 if (ar[19] == 1) Serial.print(" BattOk "); else Serial.print(" BattLow ");

 int t=0; for (int i=0; i <= 10; i++) { t=t << 1; t=t+ar[20+i];  }
 t=1647-t; Serial.println(t);  //temperature in degrees*10
}
```

Type 2 Funk-Temperatur sensor TFA 30.3120.90

Sends data first few minutes with 10 s intervals and later with 1 min intervals.

Repeats same data twice with a short pause in between. Oscillogramm for 27.4C.



Short pulse 530-570 us.
Long pulse1280-1300 us.
Pause 949 us always same length.

// Arduino code to record truth table

```
byte pin = 13; // from receiver. Between receiver and Arduino is 2 transistor 3V to TTL voltage converter
byte ar[116]; byte pos=0; unsigned long dur; //array, position in array, pulse duration

void setup() { pinMode(pin, INPUT);  Serial.begin(9600); Serial.flush(); }

void loop() { b: pos=0;
  dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 0; pos++;
   dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 0; pos++;
    dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 0; pos++;
     dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 0; pos++;
      dur = pulseIn(pin, HIGH); if ((dur>400)&&(dur<700)) {ar[pos] = 0; pos++;
      }else goto b;
     }else goto b;
    }else goto b;
   }else goto b;
  }else goto b;
 //  Serial.print("data "); // if one gets to this point then message is most probably correct
 for (int i=1; i <= 39; i++){
  dur = pulseIn(pin, HIGH);
  if ((dur>400)&&(dur<700)){ar[pos] = 0; pos++; Serial.print("0");}
  if ((dur>1100)&&(dur<1500)){ar[pos] = 1; pos++; Serial.print("1");}
  }
 Serial.println(" end ");

}
```

Truth table recorded from the type 2 wireless temperature sensor

111101011111 1100110 0 1000 1000 1011 1000 1000 1111 end 27.4
111101011111 1100110 1 1000 1000 1001 1000 1000 1110 end 27.6
111101011111 1100110 0 1000 1000 0111 1000 1000 1011 end 27.8
111101011111 1100110 0 1000 0111 0111 1000 0111 1001 end 28.8
111101011111 1100110 0 1000 0110 0110 1000 0110 0110 end 29.9

111101011111 1100110 0 0111 1110 1011 0111 1110 1001 end 31.4
111101011111 1100110 0 0111 1100 1010 0111 1100 0100 end 33.5
111101011111 1100110 0 0111 1000 1000 0111 1000 1010 end 37.7

111101011111 1100110 0 0110 1100 1101 0110 1100 0101 end 43.2
111101011111 1100110 1 0110 0111 1101 0110 0111 1100 end 48.2

111101011111 1100110 0 0101 1101 1010 0101 1101 0010 end 52.5
111101011111 1100110 0 0101 1001 1101 0101 1001 1101 end 56.2
111101011111 1100110 0 0101 0110 1110 0101 0110 1000 end 59.1

111101011111 1100110 0 0100 1110 0111 0100 1110 1111 end 61.8
111101011111 1100110 1 0100 1011 1111 0100 1011 0010 end 64.0
111101011111 1100110 1 0100 1000 1001 0100 1000 0110 end 67.6
111101011111 1100110 1 0100 0110 1110 0100 0110 0111 end 69.1
111101011111 1100110 1 1111 0000 0000 1111 0000 0011 end ofl >70C

44 bits altogather:
12 bits device type 111101011111
7 bits device ID changes every time batteries inserted
1 bit checksum ? 1 if number of ones in next 12 bits is even.
4 bits tens. Subtract 5 to get degrees.
4 bits ones
4 bits decimal parts
4 bits tens again
4 bits ones again
4 bits tens new

For single degrees and decimal parts:
1111 > invert > 0
1110 > invert > 1
1101 > invert > 2
1100 > invert > 3
1011 > invert > 4
1010 > invert > 5
1001 > invert > 6
1000 > invert > 7
0111 > invert > 8
0110 > invert > 9

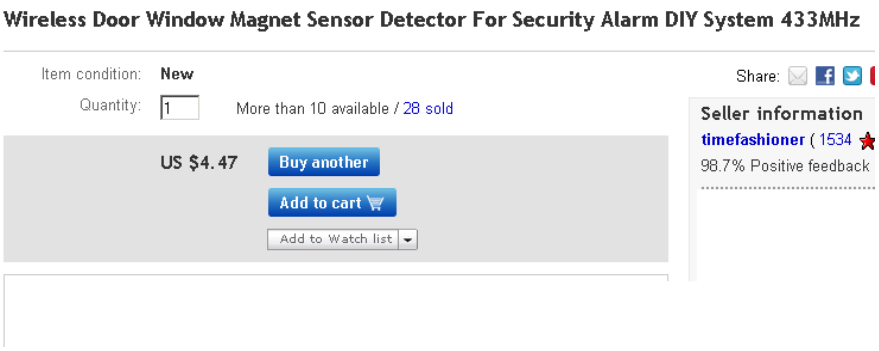
For tens of degrees invert and subtract 5

-40 1
-30 2
-20 3
-10 4
0 5
10 6
20 1000 > 0111 = 7
30 0111 > 1000 = 8
40 0110 > 1001 = 9
50 0101 > 1010 = 10
60 0100 > 1011 = 11
70 1111 > 0000 = 0

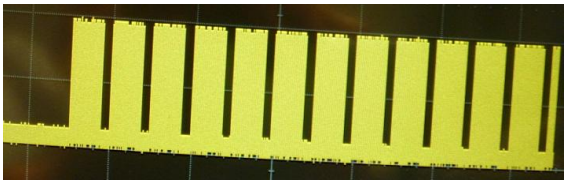
// Funksensor TFA, 433 MHz Temperatur-Außensender TFA 30.3120.90 not compatible with receiver 30.3034.01
// read and decode temperature. 44 bits alltogather

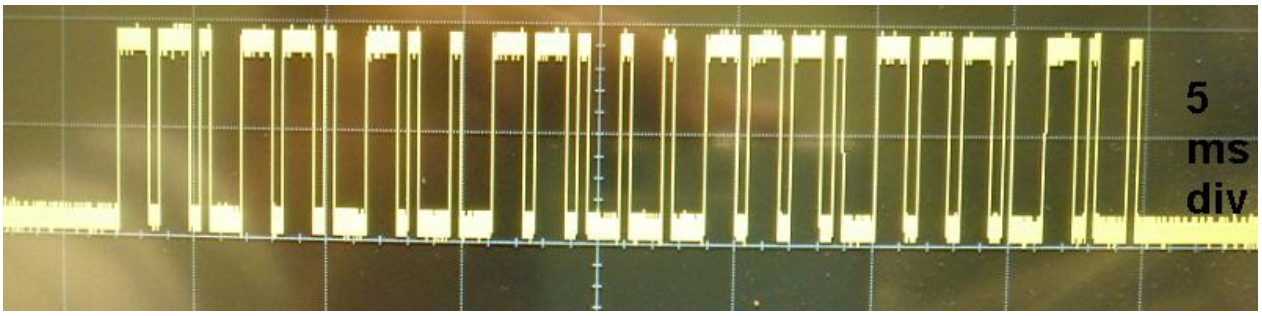
```
// 12 bits device type 111101011111
// 7 bits device ID changes every time batteries inserted
// 1 bit  checksum is one if bit sum in first 3 digits is even
// 4 bits inverted tens. Subtract 5 to get degrees. 4 bits inverted ones. 4 bits inverted decimal parts
// 4 bits invertedtens again. 4 bits invertedones again. 4 bits inverted tens. We don't use this.

byte pin = 13; // from receiver. Between receiver and Arduino is 2 transistor 3V to TTL voltage converter
byte ar[116], pos=0; unsigned long dur; //array, position in array, pulse duration
int b, d, t;
void setup() { pinMode(pin, INPUT);  Serial.begin(9600); Serial.flush(); }

void loop() { b: pos=0;
// 1111 01 011111 device type other temp sensor TFA 30.3120.90
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>400)&&(dur<700)) {ar[pos] = 0; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>400)&&(dur<700)) {ar[pos] = 0; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
dur = pulseIn(pin, HIGH); if ((dur>1100)&&(dur<1500)) {ar[pos] = 1; pos++;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
//  Serial.print("data "); // if one gets to this point then message is most probably correct

 for (int i=1; i <= 32; i++){
  dur = pulseIn(pin, HIGH);
  if ((dur>400)&&(dur<700)){ar[pos] = 0; pos++; /* Serial.print("0"); */ }
  if ((dur>1100)&&(dur<1500)){ar[pos] = 1; pos++; /* Serial.print("1"); */}
/* if (i==7) { Serial.print(" "); }
  if (i==8) { Serial.print(" "); }
  if (i==12) { Serial.print(" "); }
  if (i==16) { Serial.print(" "); }
  if (i==20) { Serial.print(" "); }
  if (i==24) { Serial.print(" "); }
  if (i==28) { Serial.print(" "); }
  if (i==32) { Serial.print(" "); } */
  }

 Serial.print("databegin TFA2 ");
  d=0; for (int i=0; i <= 6; i++) { /*if (ar[12+i]==1) Serial.print("1"); if (ar[11+i]==0) Serial.print("0");*/
  d=d << 1; d=d+ar[12+i]; }  Serial.print(d); Serial.print(" "); // Device ID

  d=0; for (int i=0; i <= 3; i++) {d=d << 1; if (ar[20+i]==1) b=0; else b=1; d=d+b;} d=d-5; t=d*100; // tens of degrees
  d=0; for (int i=0; i <= 3; i++) {d=d << 1; if (ar[24+i]==1) b=0; else b=1; d=d+b;} t=t+10*d; // ones of degrees
  d=0; for (int i=0; i <= 3; i++) {d=d << 1; if (ar[28+i]==1) b=0; else b=1; d=d+b;} t=t+d;  // decimal parts of degrees
 Serial.print(t);

  d=ar[19]; for (int i=0; i <= 11; i++) {if (ar[20+i]==1) d=d+1;} //  Serial.println(d);
  if ( (d % 2) == 0) Serial.println(" CRCerror"); else Serial.println(" CRCok");
}
```

Door switch



Many packets >10





11011010011000111011101100

25 bits in packet.

Long pulse: 3 high, 1 low
Short pulse: 1high, 3 low

1160 us 3high
1159 us 3 low
394 us 1 low
385 us 1 high

PIR motion detector

Wireless Motion PIR Infrared Sensor Detector For Security Alarm System 433MHz

Item condition: **New**
Ended: May 05, 2012 19:58:23 PDT
Winning bid: **US $8.00** [8 bids]
Add to list

BillMeLater **$5 back on first Bill Me Later purchase**
Subject to credit approval. See terms

Shipping: **FREE** - Economy Int'l Shipping | See all details
See details about international shipping here.
Item location: **Hong Kong, Hong Kong**
Ships to: **Worldwide**

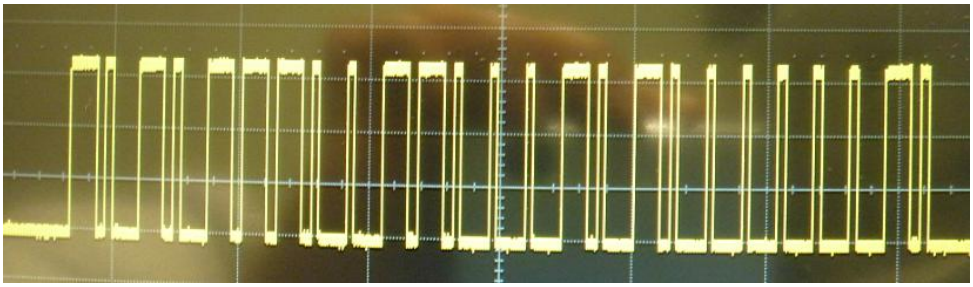
Delivery: Varies for items shipped from an international location
Seller ships within 5 days after receiving cleared payment

Payments: PayPal, Bill Me Later | See details

Returns: 14 days money back, buyer pays return shipping | Read details

Seller information
timefashioner (1534 ★)
98.7% Positive feedback

Save this seller
See other items
Visit store: timefashioner



```
1010 1110 0110 0010 1000 0001 0
```

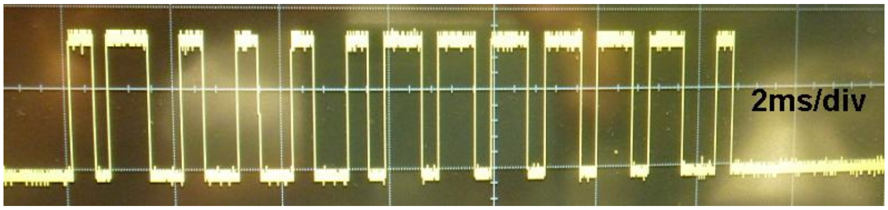
25 pulses

979-1035 us 3 high, long
279-348 us 1 high, short
1007 3 low, long pause length changes

Arduino code had to have slightly different sequence.
Don’t know why. May be the pulses are so short that Arduino misses some.

```
// receiver for 433 MHz PIR motion detector from ebay

byte pin = 13; // from receiver. Between receiver and Arduino is 2 transistor 3V to TTL voltage
converter
byte ar[116], pos=0; int dur; //array, position in array, pulse duration
int b, d, t;
void setup() { pinMode(pin, INPUT);  Serial.begin(9600); Serial.flush(); }
// 1010 1110 0110 0010 1000 0001 0
// 1010 1011 1001
void loop() { b:
dur = pulseIn(pin, HIGH); if ((dur>800)&&(dur<1300)){
dur = pulseIn(pin, HIGH); if ((dur>200)&&(dur<450)) {
dur = pulseIn(pin, HIGH); if ((dur>800)&&(dur<1300)) {
dur = pulseIn(pin, HIGH); if ((dur>200)&&(dur<450)) {
dur = pulseIn(pin, HIGH); if ((dur>800)&&(dur<1300)) {
dur = pulseIn(pin, HIGH); if ((dur>200)&&(dur<450)) {
dur = pulseIn(pin, HIGH); if ((dur>800)&&(dur<1300)) {
dur = pulseIn(pin, HIGH); if ((dur>800)&&(dur<1300)) {
dur = pulseIn(pin, HIGH); if ((dur>800)&&(dur<1300)) {
dur = pulseIn(pin, HIGH); if ((dur>200)&&(dur<450)) {
dur = pulseIn(pin, HIGH); if ((dur>200)&&(dur<450)) {
dur = pulseIn(pin, HIGH);if ((dur>800)&&(dur<1300)) {

}else goto b;
}else goto b;
}else goto b;
}else goto b;

}else goto b;
}else goto b;
}else goto b;
}else goto b;

}else goto b;
}else goto b;
}else goto b;
}else goto b;
//  Serial.print("data "); // if one gets to this point then message is most probably correct

 Serial.println("databegin PIR");
delay(1000);
}
```

Doorbell



13 bits. Every time the small button inside is pressed the code changes completely. There is no loop, always new code.

Long 685 699 us
Short 384-396 us
Pause long 630 us. Pause short 243 – 294 us

```
// receiver for 433 MHz PIR motion detector from ebay

byte pin = 13; // from receiver. Between receiver and Arduino is 2 transistor 3V to TTL voltage
converter
int dur; //array, position in array, pulse duration
void setup() { pinMode(pin, INPUT);  Serial.begin(9600); Serial.flush(); }
void loop() { b:
// 13 bits code changes every time the button inside is pressed
dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<500)) {
dur = pulseIn(pin, HIGH); if ((dur>501)&&(dur<701)) {
dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<500)) {
dur = pulseIn(pin, HIGH); if ((dur>501)&&(dur<701)) {
dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<500)) {
dur = pulseIn(pin, HIGH); if ((dur>501)&&(dur<701)) {
dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<500)) {
dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<500)) {
dur = pulseIn(pin, HIGH); if ((dur>501)&&(dur<701)) {
dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<500)) {
dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<500)) {
dur = pulseIn(pin, HIGH); if ((dur>501)&&(dur<701)) {
dur = pulseIn(pin, HIGH); if ((dur>300)&&(dur<500)) {
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;
}else goto b;

Serial.println("databegin bell");
delay(2000);
}
//Next step will be to write program that unites all sensors
```