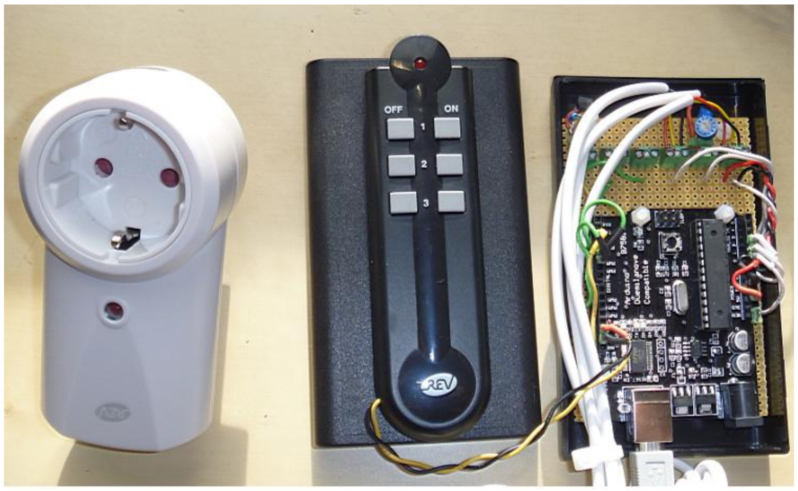
Wireless power outlets for home automation using Arduino

Home automation becomes more and more popular, affordable and fascinates people. Internet offers such possibilities as never before. Impress your friends showing on Iphone that you can switch on/off lamp in your apartment 1000 km away and simultaneously see it through a webcam.

Wirelessly controlled power outlets are best suited for switching on/off appliances as they do not need cables and only the remote control needs to be interfaced to a microcontroller. Design is electric -shock -safe as the high voltage modules are not opened.

Arduino microcontroller greatly simplifies the task, because it is a standard board that can be easily reprogrammed. Arduino could be connected to a PC, WiFi router with USB running OpenWRT, or direct connection to Internet could be done via Arduino Ethernet shield. Picture below illustrates how home automation box could look. There are external sensors attached to Arduino, PIR, photodiode, sound level monitor, 1-wire temperature sensors DS18B20.



Before building one please prepare yourself by reading through these links:

Arduino Homecontrol 4 me. http://www.homecontrol4.me/de/?id=bauanleitung
Very professional. Not only sends but also receives 433 MHz. Could have some wireless temperature sensors, switches motion detectors. Quite hard coding, need to flash custom EEPROM.

For AVR microcontrollers. Funksteckdosen mit AVR steuern http://www.mikrocontroller.net/topic/131435

Almoust Arduino, but simpler. One chip, direct USB connection, charge pump to make 12 V for transmitter. But more difficult to reprogram than Arduino.
http://www.instructables.com/id/USB-controlled-home-automation-hack/?ALLSTEPS

10 EUR Funk – Schalter set 6899 433 MHz. Removed from market because of fire danger
http://avr.börke.de/Funksteckdosen.htm
http://www.baua.de/de/Publikationen/BAuA-AKTUELL/2007-3/pdf/ba3-07-s09.pdf?__blob=publicationFile&v=2

Conrad.de 10 EUR only one remote and one outlet. Or 20 EUR 3. Design looks crappy.
http://blog.sui.li/2011/04/12/low-cost-funksteckdosen-arduino/

Code for Arduino nicely written
http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1294088583

Hardware

Best choise found at the moment of writing:

Funksteckdose
REV Telecontrol 1000 W 1+3 er Set' von www.rev.biz
Art. Nr 008345 <20 EUR. For this price can not make self.

35 m, 433 MHz, nice looking design.

Remote control has 4 switch selections A, B, C, D and 3 ON/OFF buttons allowingv to control 12 loads.

Power outlets are quite smart. They remember the last state also when removed from power.

Disadvantage - red LED on all time, does not indicate status of load.

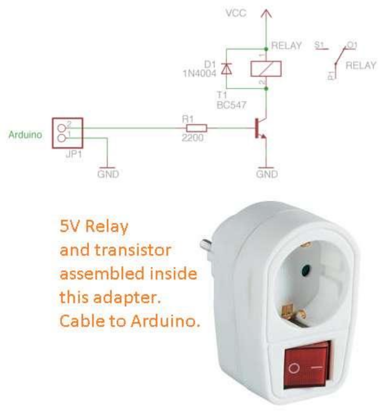
Power consumption measured of one outlet was below 1W. The meter showed 0.0W.

Chip gets power less than the battery voltage, ca 5V. The signal amplitude from chip is TTL.

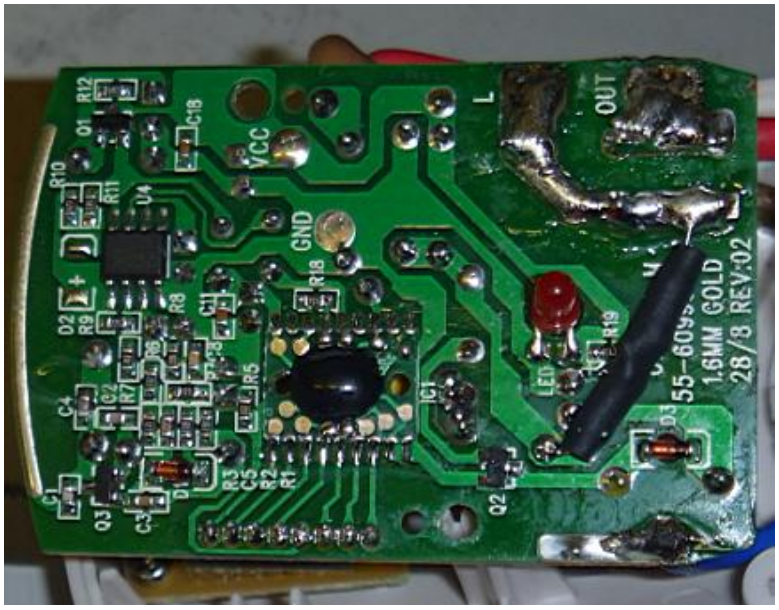
Smart buttons on circuit board. Double pressed layout. Power connected to chip and also to button encoder.

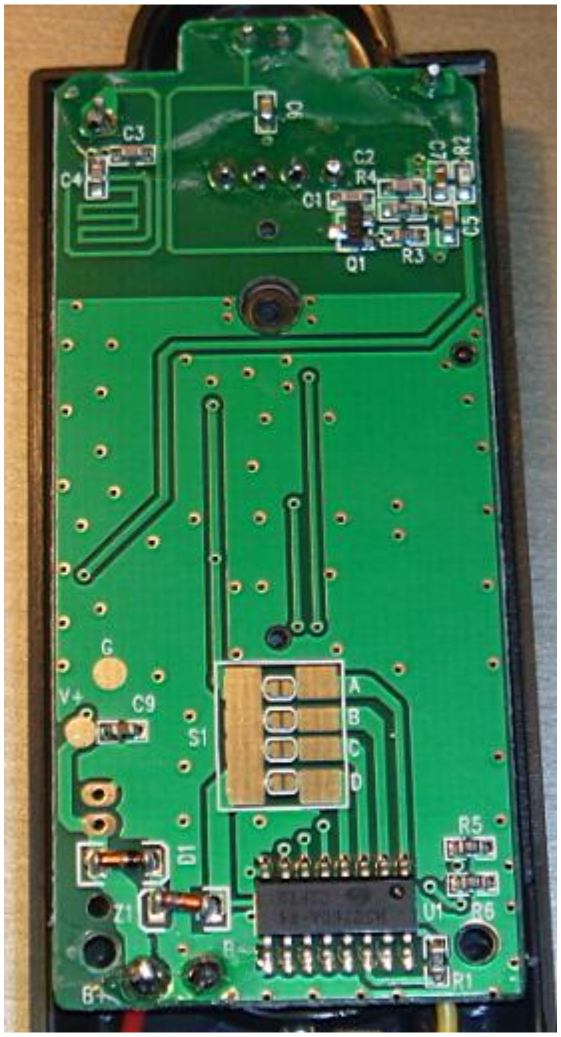
SAW filter 433 MHz as resonator for transmitter. Good thermal stability.

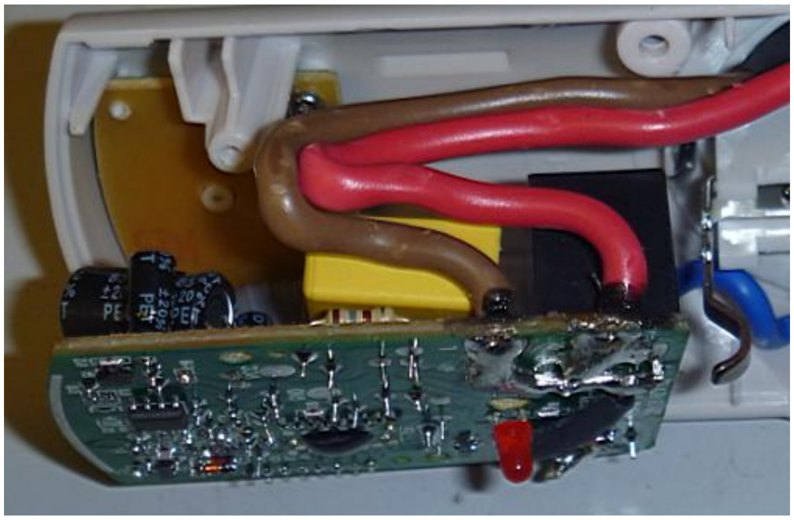
My previous solution to build in a relay into an outlet is not CE certified and probably illegal,
but wireless Funksteckdose remains certified and thus it could be even sold.

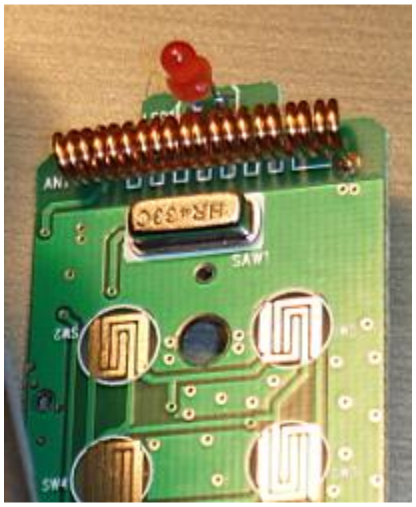


Reverse engineering



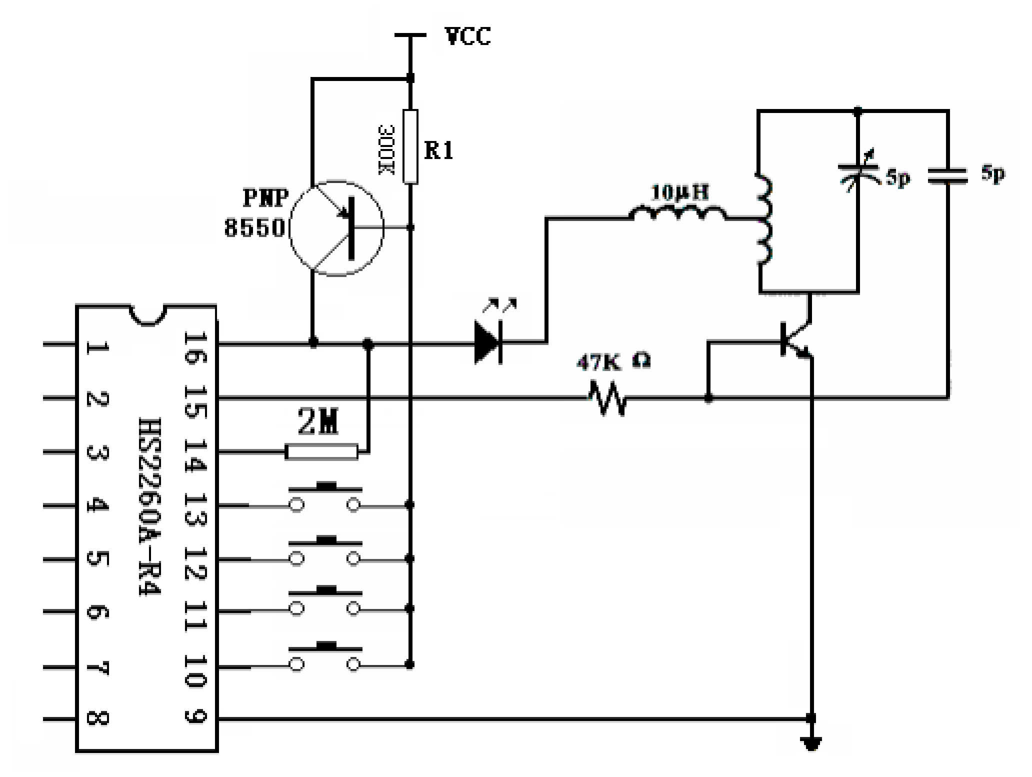


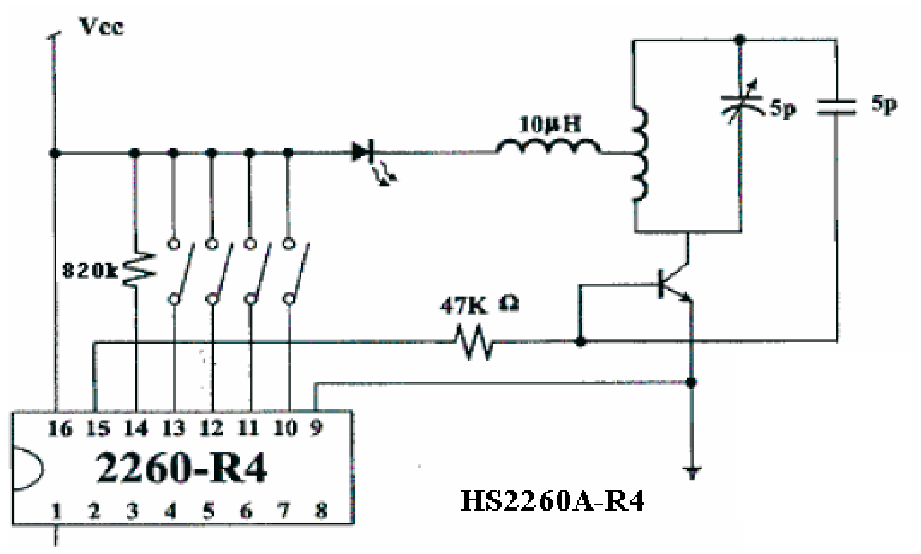


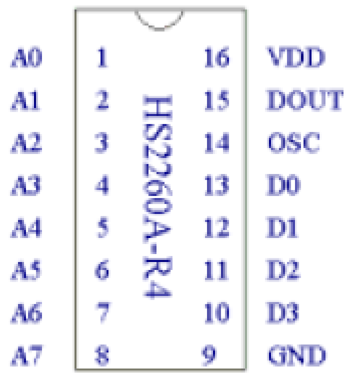


Schematics

Circuit below is not exactly what is inside. The transmitter actually has a 433 MHz SAW oscillator.

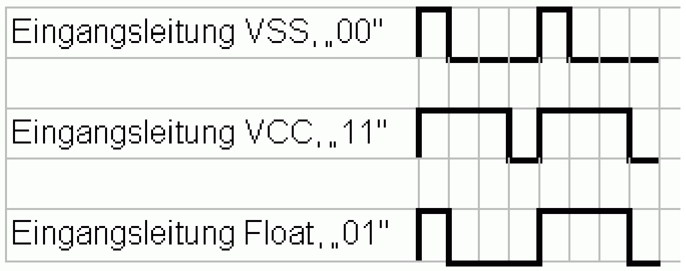


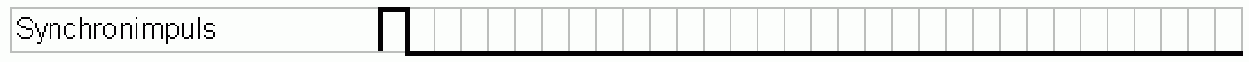




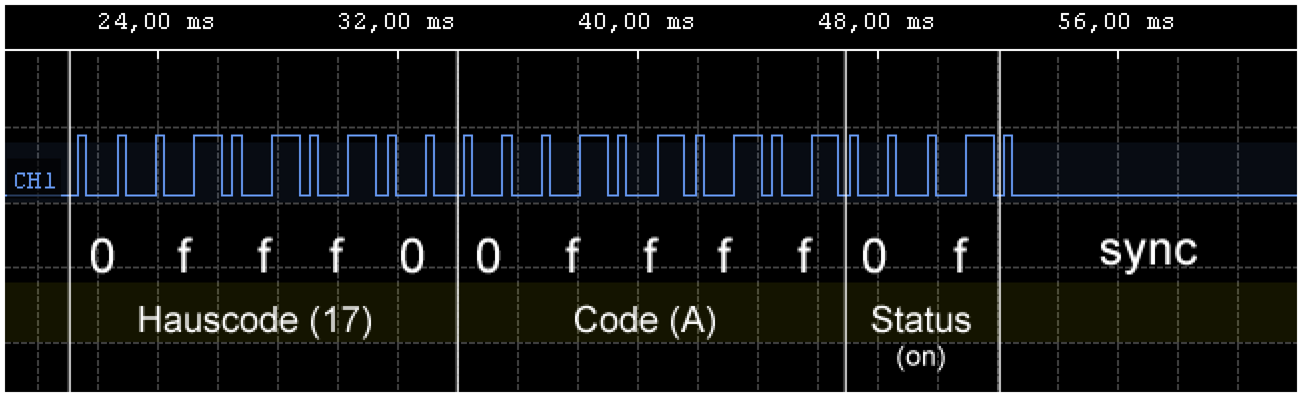
Signals

ASK modulation. Amplitude modulation. Three basic signals:

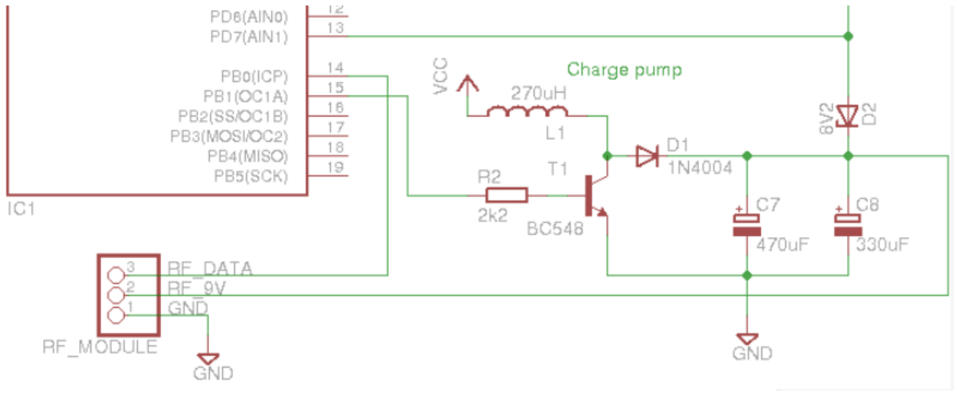

Eingangsleitung VSS, „00"
Eingangsleitung VCC, „11"
Eingangsleitung Float, „01"



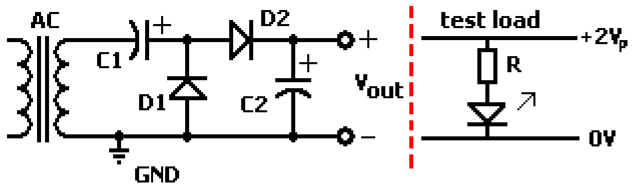
http://www.maltepoeggel.de/?site=usbfunk&p=3&lang=de



Remote needs 12 V, originally supplied by a battery. If you don't want a battery in your gadget then can make a charge pump from Arduino.
http://www.instructables.com/id/USB-controlled-home-automation-hack/?ALLSTEPS



This kind of voltage doubler should work too on Arduino PWM output:



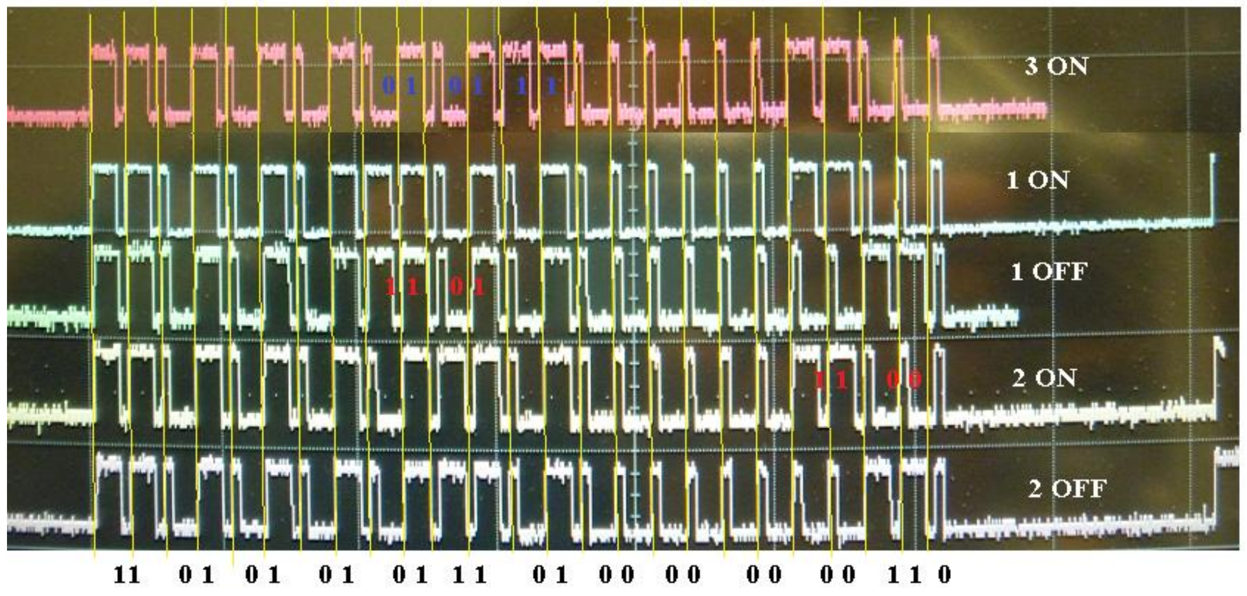
Code for Arduino nicely written http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1294088583
Basics for one button. For hardware described here http://avr.börke.de/Funksteckdosen.htm

```
int fdelay = 342;
int funkpin = 2;  //Anschluß an FB (Data Ausgang HX2262)
int repeat = 8;

char* code = "123456789012";

void setupFunk(){
 pinMode(funkpin, OUTPUT);
}
void loopFunk(){
}
void codeSenden(char* sc){
 char ch;

 for(int i=0; i<=repeat; i++){
   for(int j=0; j<=11; j++){
     ch = c[j];
     switch (ch){
       case "0":
         digitalWrite(funkpin, HIGH);
       delayMicroseconds(fdelay);
       digitalWrite(funkpin, LOW);
       delayMicroseconds(fdelay*3);
       digitalWrite(funkpin, HIGH);
       delayMicroseconds(fdelay);
       digitalWrite(funkpin, LOW);
       delayMicroseconds(fdelay*3);
       break;
       case "F":
         digitalWrite(funkpin, HIGH);
       delayMicroseconds(fdelay);
       digitalWrite(funkpin, LOW);
       delayMicroseconds(fdelay*3);
       digitalWrite(funkpin, HIGH);
       delayMicroseconds(fdelay*3);
       digitalWrite(funkpin, LOW);
       delayMicroseconds(fdelay);
         break;
       case "1":
         digitalWrite(funkpin, HIGH);
       delayMicroseconds(fdelay*3);
       digitalWrite(funkpin, LOW);
       delayMicroseconds(fdelay);
       digitalWrite(funkpin, HIGH);
       delayMicroseconds(fdelay*3);
       digitalWrite(funkpin, LOW);
       delayMicroseconds(fdelay);
         break;
     }
   }
   //Sende Sync 32 Takte
   digitalWrite(funkpin, HIGH);
   delayMicroseconds(fdelay);
   digitalWrite(funkpin, LOW);
   delayMicroseconds(fdelay*31);
 }
}
```

Oscilloscope signals

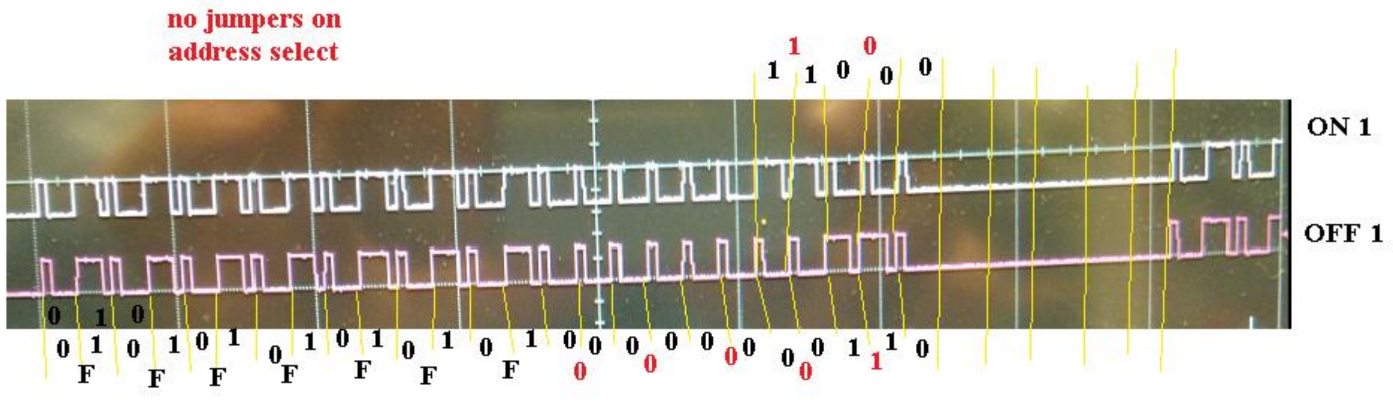
Measured on the chip output pin 15 using digital oscilloscope. Buttons for switch A only.

Vert. scale 5V/div. Pulses repeat every 1.28 ms.



Haven't tried to decipher encoding. Simply Arduino should reproduce these signals. Can connect Arduino to pin 15 of the remote control chip. No level matching was necessary as the signal goes to the transistor base of the RF oscillator via a resistor.

Additional signals recorded when ABCD jumper was removed.



Further things to do: wireless temperature sensrors, weather station, PIR motion detector door/window switches.
http://www.practicalarduino.com/projects/weather-station-receiver
https://github.com/practicalarduino/WeatherStationReceiver
http://www.practicalarduino.com/about
Perl sensor lesen
http://www.mikrocontroller.net/topic/252922

My code

Copy - paste into Arduino SDK. Use Arduino0022 to compile. Code worked for me, sorry somewhat ugly.
Check with digital oscilloscope that Arduino produces signals identical to original remote.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define TEMPERATURE_PRECISION 12
#define ONE_WIRE_BUS 2 // Onewire data wire is plugged into port 2 on the Arduino. Needs ca 4k pullup resistor to 5V.

OneWire oneWire(ONE_WIRE_BUS); // Setup 1Wire instance to communicate with any OneWire devices (not just Maxim/Dallas temperature ICs)
DallasTemperature sensors(&oneWire); // Pass our oneWire reference to Dallas Temperature.
DeviceAddress insideThermometer, outsideThermometer, thirdThermometer; // arrays to hold device addresses
int d=314;  // delay adjusted to macth original wireless remote.  Use a digital oscilloscope to check.
int tpin=13;  // 433 MHz RF transmitter hooked to this pin (soldered parallel to the remote encoding chip output.)
int dig1pin=11; int dig2pin=12; // digital outputs that can be switched on with a serial command
String dig1="d1off "; String dig2="d2off ";
String tA1=" A1? ";  String tA2="A2? "; String tA3="A3? "; // last command output to wireless socket
int count=0; //byte count in serial message

void setup() {
  pinMode(tpin, INPUT); // we unblock transmit pin when not in use so that normal remote functions too.
  pinMode(dig1pin, OUTPUT);
  pinMode(dig2pin, OUTPUT);

  Serial.begin(9600); Serial.flush();
  Serial.println ("Reads AI0, AI1, AI2. Controls digital output pins 11, 12. Commands to switch digital outputs: D1on, D1off, D2on, D2off.");
  Serial.println ("Generates 1-wire bus (pin2) for DS18B20 temperature sensors. Needs external4.7k pullup.");
  Serial.println ("433 MHz wireless power outlet chip HS2260A-R4 simulation (pin 13). Commands: A1on, A1off, A2on, A2off, A3on, A3off.");
  Serial.println ("Reset button of Arduino is disabled to 5V, remove jumper for reprogramming");
  Serial.println ("Make sure there is no echo from USB host back to Arduino, buffer will overload and commanding will not work");
   sensors.begin();
    if (!sensors.getAddress(insideThermometer, 0)) Serial.println("Unable to find  1-wire sensor nr 1");
    if (!sensors.getAddress(outsideThermometer, 1)) Serial.println("Unable to find 1-wire sensor nr 2");
    if (!sensors.getAddress(thirdThermometer, 2)) Serial.println("Unable to find 1-wire sensor nr 3");
    sensors.setResolution(insideThermometer, TEMPERATURE_PRECISION);
    sensors.setResolution(outsideThermometer, TEMPERATURE_PRECISION);
}
// functions for wireless remote control emulation
  void I() {     // sends one
          digitalWrite(tpin, HIGH); delayMicroseconds(3*d);   digitalWrite(tpin, LOW);  delayMicroseconds(1*d);
          digitalWrite(tpin, HIGH); delayMicroseconds(3*d);   digitalWrite(tpin, LOW);  delayMicroseconds(1*d); }
  void O() {    // sends zero
          digitalWrite(tpin, HIGH); delayMicroseconds(1*d);   digitalWrite(tpin, LOW);  delayMicroseconds(3*d);
          digitalWrite(tpin, HIGH); delayMicroseconds(1*d);   digitalWrite(tpin, LOW);  delayMicroseconds(3*d); }
  void F() {   // sends float
          digitalWrite(tpin, HIGH); delayMicroseconds(1*d);  digitalWrite(tpin, LOW);  delayMicroseconds(3*d);
          digitalWrite(tpin, HIGH); delayMicroseconds(3*d);  digitalWrite(tpin, LOW);  delayMicroseconds(1*d); }
  void E()  {   // synchronization
         digitalWrite(tpin, HIGH);  delayMicroseconds(1*d);  digitalWrite(tpin, LOW);   delayMicroseconds(31*d);  }

// functions for 1wire bus
void printAddress(DeviceAddress deviceAddress) { for (uint8_t i = 0; i < 8; i++)  {  }  }
void printTemperature(DeviceAddress deviceAddress) { float tempC = sensors.getTempC(deviceAddress); Serial.print(tempC); Serial.print(" ");  }
void printResolution(DeviceAddress deviceAddress) {  }
void printData(DeviceAddress deviceAddress) {  printAddress(deviceAddress);  printTemperature(deviceAddress); }

void loop() {
String kommand=""; // string received via com port
String report=""; // string output to com port
count=Serial.available();
 for (int x=0; x<count; x++) { char ch = Serial.read(); kommand = kommand + ch;}
//if (kommand != "") {Serial.print (kommand); Serial.println (count);}
```

```
     kommand=kommand.substring(0,4);
  if (kommand=="A1on") {tA1=" A1on ";   pinMode(tpin, OUTPUT);
    for (int i=0; i <= 20; i++) { I();F();F();F();I();F();F();O();O();O();I();O();E(); }    }

  if (kommand=="A1of") {tA1=" A1off "; pinMode(tpin, OUTPUT);
    for (int i=0; i <= 20; i++) { I();F();F();F();I();F();F();O();O();O();O();I();E(); }    }

  if (kommand=="A2on") {tA2="A2on "; pinMode(tpin, OUTPUT);
    for (int i=0; i <= 20; i++) { I();F();F();F();F();I();F();O();O();O();I();O();E(); }    }

  if (kommand=="A2of") {tA2="A2off "; pinMode(tpin, OUTPUT);
    for (int i=0; i <= 20; i++) { I();F();F();F();F();I();F();O();O();O();O();I();E(); }    }

  if (kommand=="A3on") {tA3="A3on "; pinMode(tpin, OUTPUT);
    for (int i=0; i <= 20; i++) { I();F();F();F();F();F();I();O();O();O();I();O();E(); }    }

  if (kommand=="A3of") {tA3="A3off "; pinMode(tpin, OUTPUT);
    for (int i=0; i <= 20; i++) { I();F();F();F();F();F();I();O();O();O();O();I();E(); }     }

  if (kommand=="d1on") {dig1="d1on "; digitalWrite(dig1pin, 1);}
  if (kommand=="d1of") {dig1="d1off "; digitalWrite(dig1pin, 0);}
  if (kommand=="d2on") {dig2="d2on "; digitalWrite(dig2pin, 1);}
  if (kommand=="d2of") {dig2="d2off "; digitalWrite(dig2pin, 0);}
pinMode(tpin, INPUT);

sensors.requestTemperatures();  printData(insideThermometer);  printData(outsideThermometer);  printData(thirdThermometer);
report = report+analogRead(A0)+" "+analogRead(A1)+" "+analogRead(A2)+" "+tA1+tA2+tA3+dig1+dig2;
Serial.println(report);
delay(500);
}
```

Read out and control for OpenWRT Linux

Need Arduino Duemillanovwe with FTDI serial chip.
Use progrom stty to set serial speed.

Important!!
Disable echo back to Arduino, its input buffer will fill up and Arduino will not listen to commands/

```
opkg update
opkg install kmod-usb2
opkg install coreutils-stty
opkg install kmod-usb-serial
opkg install kmod-usb-serial-ftdi
```

A device should appear with plugged in Arduino
ls /dev/ttyUSB0

```
#!/bin/sh
#/var/www/cgi-bin/readuino.sh

# simple script to tail newline terminated messages from the arduino
# adjust the head number depending on your arduino messages

if  ! [ -f /tmp/isrunning ]; then
if  [ -c /dev/ttyUSB0 ]; then

echo $$ > /tmp/isrunning
/usr/bin/stty -F /dev/ttyUSB0  speed 9600 -echo
while [ 1 ]; do
if [ -c /dev/ttyUSB0 ]; then
head -n 1 /dev/ttyUSB0 >/tmp/volts1
cp /tmp/volts1 /tmp/volts
sleep 1
else
# USB dissapeared.
echo "Arduino no longer attached" >/tmp/volts
rm /tmp/isrunning
exit
fi
done
else
echo "Arduino not yet  attached" >/tmp/volts
fi
fi
```

```
#!/bin/sh
#!/www/cgi-bin/arduino
# simple CGI to tail most recent info from an arduino

#opkg update
#opkg install coreutils-stty
#opkg install kmod-usb-serial
#opkg install kmod-usb-serial-ftdi

if ! [ -f /tmp/isrunning ]; then
/www/cgi-bin/readuino.sh & 2>&1 >/dev/null
fi

echo "Content-type: text/html"
echo ""
echo "<html><head><meta http-equiv="refresh" content="2"></head><body>"
echo "<font color=blue><h2>Pocket router TP-Link WR703N interface to Arduino microcontroller board</h2>"
echo "Raw data received from Arduino /dev/ttyUSB0:<br>"
a=""
a=$(cat /tmp/volts)
echo $a
set -- $a
t1=$1
t2=$2
t3=$3
ai0=$4
ai1=$5
ai2=$6
A1=$7
D1=$10
echo "<font color=red><hr><b>Thermometers DS18B20 on 1-wire bus, Arduino pin 2</b>"
echo "<br> Temperature inside Arduino: " $t1
echo "<br> Temperature outside: " $t2
echo "<br> Temperature, for example, heating pipe: " $t3
echo "<hr><b><font color=green>Arduino 10 bit analog inputs ai0, ai2, ai2</b> "
echo "<br> PIR motion detector: " $ai0
echo "<br> Light intensity" $ai1
echo "<br> Sound level: " $ai2
echo "<hr> Arduino digital output pin 11: " $D1
echo "<form><input type='button' value='Switch ON ' onClick='window.location=3'></form>"
echo "<form><input type='button' value='Switch OFF' onClick='window.location=2'></form>"

echo "<hr><b><font color=brown>Wireless wallplug outlet 433 MHz transmitter on Arduino pin 13, wallplug address A1.</b>"
echo "<br> Mains power: " $A1
echo "<form><input type='button' value='Switch ON ' onClick='window.location=1'></form>"
echo "<form><input type='button' value='Switch OFF' onClick='window.location=0'></form>"
echo "<hr></html>"
echo ""
```

```
 #!/bin/sh
#!/www/cgi-bin/1
# switches on 432 MHz power outlet
echo "Content-type: text/html"
echo ""
echo "<html><body>"

if  [ -c /dev/ttyUSB0 ]; then
echo "/dev/USB0 exists, OK <br>sending command 1 ON"

echo "A1on">/dev/ttyUSB0
else
echo "/dev/ttyUSB0 does not exist<br> can not send command"

fi
echo "</body></html>"
```
