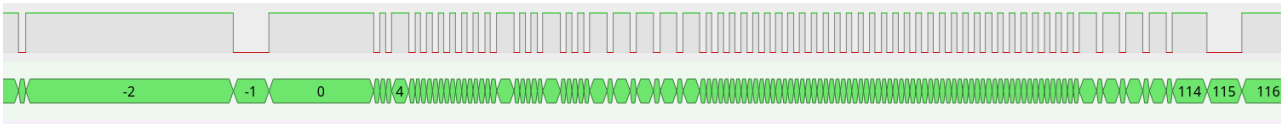
Řízení klimatizace Samsung

Klimatizace používá k řízení dálkové ovladače. Prvním úkolem bylo získat protokol. Bohužel protokol je proprietární a není na internetu jsem nenašel konkrétní řešení pro tuto jednotku. Cílem bylo postavit zařízení, které by do vstupu dálkového ovládání pouštělo stejný signál, jako by byl stisk na dálkovém ovladači. Dálkový ovladač AR-EH03E se liší od ostatních ovladačů nejen protokolem, ale i tím, že je stavový – tedy dálkový ovladač si uchovává stav a zobrazuje ho na displeji.

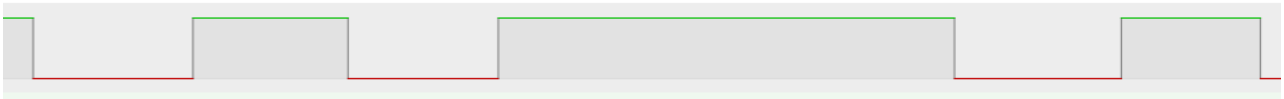
Dekódování DO AR-EH03E

Osciloskopem jsem zjistil, že DO vysílá na 38 kHz. Použil jsem integrovaný přijímač SFH5110-38 pro dekódování a logický analyzátor. Výsledkem byla sekvence začínající krátkým pulsem, pak 18 ms nic a pak další puls 3ms. Pak pauza 9ms a pak data 7 bytů při kódování 0 krátký puls, 1 dlouhý puls. Na konci je ještě 1 bit navíc, ale s délkou nuly 3ms. Pak následuje další rámec s 3 ms pulsem. Zajímavé je, že bit navíc není u posledního rámce.



Rámec se začátkem

Vždy krátká jednička (500us) a krátká nula (500us) dává logickou nulu nebo krátká jednička (500us) a dlouhá nula (1500us) dává logickou jedničku.



Ukázka 0 – 1 – 0

Rámce mohou být 2 nebo 3. 3 se typicky používají pro zapnutí a vypnutí, kdežto 2 při změně, když je ovladač aktivní. Pro dekódování jsem výstup z IR přijímače zapojil na INT0 mikropočítače (Jednoduchý MCU, PIC18F47K40, zrovna byl k dispozici) a výstup UART převedl přes převodník UART-USB do počítače. MCU posílá textové řetězce dekódovaných dat, které jsem zobrazil v terminálu.

Pro začátek bylo potřeba určit rámce. Rámec 1 nese informaci o zapnutí/vypnutí a případně další rozšiřující informace. Pak následuje rámec 2, který je přítomen jen při zapnutí a vypnutí a dálkový ovladač zde neposílá žádné užitečné informace (ať je nastavení jakékoliv, vždy je stejný). Poté následuje rámec 3, který nese informaci o nastavené teplotě.

```
HDR1         CHK1              DI                Q
01000000 01001001 11110000 00000000 00000000 00000000 00001111
```

Příklad rámce 1

```
HDR3     ZoneCHK2 P  SWING  TULL  C YC  Temp Fan+Mode  RB pp
10000000 01000111 01111111 10001110 00000001 10000010 00000111
```

Příklad rámce 3

CHK se mění při temp i option, takže checksum, ale jen pro daný rámec
DI – direction 0 není, 1 direct, 2 indirect

Y – cycle
C – Clean
L - /Ligtning pokud je lightning, je 0
Q – Quiet
B – Beep
R – Reset
P – Pokud je příkaz vypnutí, musí být 1, jinak je 0
p – Pokud je příkaz vypnutí, musí být 0, jinak je 1
TUL – Turbo 110 / Long 011

Parametry jsem určil experimentálně měřením. Například teplota je nestandardně kódovaná podle tabulky:

```
const uint8_t samsung_temp_table[15] = {
    0x00, // 16 °C
    0x08, // 17 °C
    0x04, // 18 °C
    0x0C, // 19 °C
    0x02, // 20 °C
    0x0A, // 21 °C
    0x06, // 22 °C
    0x0E, // 23 °C
    0x01, // 24 °C
    0x09, // 25 °C
    0x05, // 26 °C
    0x0D, // 27 °C
    0x03, // 28 °C
    0x0B, // 29 °C
    0x07  // 30 °C
};
```

Nejsložitější bylo zjistit kontrolní součet. Ten počítá počet jedniček, nezapočítává vyšší nibble bytu 2 a používá nestandardní kódování, naštěstí obdobné pro počítání teploty.

```
const uint8_t Check_LUT[16] =
{
  0xF,0x7,0xB,0x3,0xD,0x5,0x9,0x1,0xE,0x6,0xA,0x2,0xC,0x4,0x8,0x0
};

uint8_t SpocitejPocetJednicekB(uint8_t Cislo)
{
  uint8_t Res=0;

  while(Cislo)
  {
    if(Cislo & 0x01) Res++;
    Cislo>>=1;
  }
  return Res;
}

// Vypočítá 4bitový CHK pro blok (stejné počítání jako při dekódování)
uint8_t Compute_Checksum(const uint8_t *b)
{
    uint8_t k = 0;
    k += SpocitejPocetJednicekB(b[0]);
    k += SpocitejPocetJednicekB(b[1] & 0xF0);  // horní nibble
    k += SpocitejPocetJednicekB(b[2] & 0x0F);  // dolní nibble
    k += SpocitejPocetJednicekB(b[3]);
    k += SpocitejPocetJednicekB(b[4]);
    k += SpocitejPocetJednicekB(b[5]);
    k += SpocitejPocetJednicekB(b[6]);

    return Check_LUT[k & 0x0F];
}
```

Takže pak jsem stanovil struktury pro práci s daty:

```
typedef union {
    uint8_t raw[7];
    struct {
        uint8_t hdr;          // 0x40 = Block 1
        uint8_t CHK:4;        // Checksum
        uint8_t unused1:4;    // Musí být 0b0100
        uint8_t unused2;      // Musí být 0xF0
        uint8_t unused3:2;    // Musí být 0b00
        uint8_t Direction:2;  // Určuje směr 00 - žádný, 01 - direct, 10 - indirect
        uint8_t unused4:4;    // Musí být 0b0000
        uint8_t unused5;      // Musí být 00
        uint8_t unused6:2;    // Musí být 0b00
        uint8_t Quiet:1;      // Aktivní Quiet
        uint8_t unused7:5;    // Musí být 0b00000
        uint8_t pwr;          // Je buďto 0x0F pro zapnutí nebo 0x03 pro vypnutí
    } data;
} SamsungAcBlock1_t;

typedef union {
    uint8_t raw[7];
    struct {
        uint8_t hdr;          // 0x80 = Block 2
        uint8_t CHK:4;        // Checksum
        uint8_t unused1:4;    // Musí být 0b0100
        uint8_t unused2;      // Musí být 0xF0
        uint8_t unused3;      // Musí být 0x00
        uint8_t unused4;      // Musí být 0x00
        uint8_t unused5;      // Musí být 0x00
        uint8_t unused6;      // Musí být 0x00 - tím odlišíme od bloku 3
    } bits;
} SamsungAcBlock2_t;

typedef union {
    uint8_t raw[7];  // 7 bajtů bloku 3

    struct {
        uint8_t block_id;     // 0x80 = Block3

        uint8_t Zone_CHK;     // Byte1 – kód pro zónu + chcecksum

        uint8_t Swing;        // Byte2 – swing kód (0x7F, 0x75, 0x73, 0xF5…)

        uint8_t Clean1:1;     // Příkaz Clean
        uint8_t unused1:2;
        uint8_t nLightning:1; // Příkaz Lightning, normálně 1, když je lightning je 0
        uint8_t Turbo:3;      // Když je Turbo je 0b110
        uint8_t unused3:1;

        uint8_t temp:4;       // Kód teploty (dekóduje se tabulkou)
        uint8_t unused4:2;
        uint8_t Clean2:1;     // Příkaz Clean
        uint8_t Cycle:1;      // Aktivní Cycle

        uint8_t modefan;      // Byte5 – mód + nastavení ventilátoru

        uint8_t unused5:5;    // Má být vždy 0b01111
        uint8_t Beep:1;       // Příkaz Beep
        uint8_t Reset:1;      // Příkaz Reset
        uint8_t unused6:1;    // Má být vždy 0
    } data;
} SamsungAcBlock3_t;
```

Po doplnění funkcí pro dekódování parametrů a jejich zobrazení, mi už MCU nejen vypisuje obsah rámců, ale počítá i kontrolní součet a vypisuje informace o stavu (co posílá DO) v čitelné podobě (tedy nastavenou teplotu, rychlost ventilátoru, mód, volby a swing).

Emulace dálkového ovladače

Když už jsem uměl dekódování, bylo možno sestavit i enkódování, tedy sestavení celého paketu. K tomu souží funkce CreatePaket, co vytvoří 14 nebo 21 bytů. Totiž při zapnutí a vypnutí se vysílá 3x7 bytů dat, v ostatních případech jen 2x7 bytů. CreatePaket sestaví paket a vrací délku dat.

Pro vysílání dat je použita funkce TransmittPacket, která sestaví okamžiky změny výstupu (časové sekvence). Poté nastaví výstup do nuly a o samotné vysílání se již stará přerušení od časovače. Tím je dosaženo přesných časů, které nejsou ovlivněny chodem programu. Funkce také neblokuje program, což je důležité pro sběrnici iDům.

Protože je výstup optočlenu napojen na SFH5110 v klimatizační jednotce, není nutno vytvářet modulovaný 38 kHz signál. Při testech v kontaktním poli jsem to sice zkoušel, ale dosah takto vyrobeného dálkového ovladače byl velmi slabý. Musel jsem přijít až 1,5 m ke klimatizaci, když to konečně začalo fungovat. V originálním ovladači asi LED opravdu přebuzují.

Hardware

Hardware je poměrně jednoduchý. Aby se klimatizační jednotka neovlivňovala se systémem iDům, je použito výstupních optočlenů. Výhodou je také, že výstup optočlenu mohu napojit paralelně k SFH5110. Druhý optočlen jsem použil pro spínání ON/OFF. Klimatizace dále obsahuje relé, které je sepnuto, když je klimatizace zapnuta. To vyhodnocuji pomocí spínání napětí. Linka iDům je řešena standardně pro PIC18F27K40, tedy N-MOSFET a rezistor. Komparátor a referenční napětí jsou součástí MCU.

iDům

Zmínil jsem se několikrát o sběrnici iDům. To je otevřená (open source) sběrnice, kterou jsem před 20 léty navrhnul pro chytré domácnosti. Cílem bylo, aby byly koncové jednotky (např. vypínače) co nejlevnější a byl umožněn režim spánku. Je určena do rodinných domů. Hardwarově se jedná o sběrnici s otevřeným kolektorem a pull-up rezistorem. Jediným problémem se ukázala kapacita sběrnice. Když to bylo v rodinném domku, tak to fungovalo výborně, ale když to bylo vymyšleno a obchoďáci se rozhodli to použít do několika propojených paneláků o 14 patrech, už začal být problém s nabíjením sběrnice. Proto jsem pull-up rezistor nahradil stavovým zdrojem proudu, který řeší právě to nabíjení sběrnice. Setkal jsem se i s rušením, kdy propojili několik vchodů přes půdu a vytvořili tak přijímací anténu pro GSM vysílač na protější střeše. To v praxi nevymyslíš. Po několik let jsem vylepšoval i protokol, který obsahuje budící sekvenci, protikolizní ochranu (zamezení kolizím) a odpovědi v náhodných časech pro vyhnutí se kolizím. S touto sběrnicí se lze setkat také v systémech domácích telefonů Czechphone.

Klimatizační jednotka

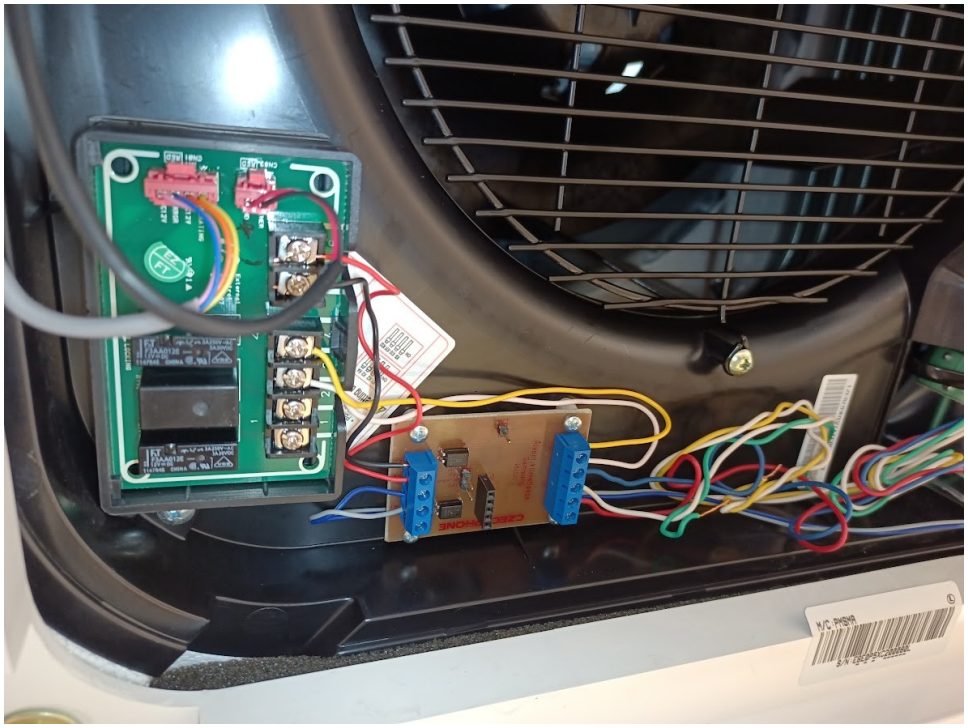
Po rozebrání je zde konektor, kde jsou zapojeny:

Oranžová	Vypínač
Hnědá	Společná zem
Fialová	Zapnuto LED anoda
Modrá	Zapnuto LED katoda
Šedá	Klapka LED anoda
Zelená	Klapka LED katoda

Fialová	Časovač LED anoda
Zelená	Časovač LED katoda
Fialová	Větrák LED anoda
Žlutá	Větrák LED katoda
Šedá	Mřížka LED anoda
Modrá	Mřížka LED katoda
Černá	+
Červená	Infra čidlo

Zatím máme bezpotenciálový spínač (s relátkem) mezi vypínač a společnou zem. Dále sledujeme svit Zapnuto LED přes optočlen. To chceme nahradit simulací dálkového ovládání na místo Infra čidla. Čidlo je klasické SFH5110, takže můžeme zapojit optočlen mezi infra čidlo a zem. Sledování a zapínání je vhodné mít také, proto je hardware navržen tak že i tyto původní funkce podporuje.

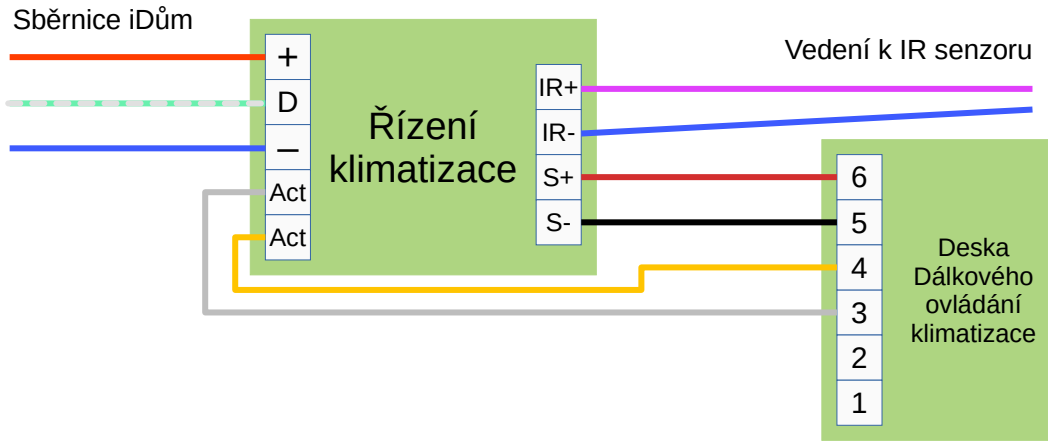
Dále je zde konektor pro napojení dálkového ovládání. To umožňuje připojit vypínač pro ON/OFF a obsahuje taky relé pro řízení stavu. A ještě jedno relé neznámého účelu.



Vývody na rozšiřující desce 6 (+) a 5 (-) jsou pro připojení bezpotenciálového spínače dálkového zapnutí/vypnutí. K sepnutí postačí i optočlen. Vývody 4 a 5 jsou napojeny na kontakty relé, které je sepnuto, když je klimatizace zapnuta. Vývody 1 a 2 jsou připojeny k relé, o kterém nevím, kdy je sepnuto.

K rozhraní tedy připojuji 6 na S+, 5 na S-, a 4 a 5 na vývody Act. Z infračerveného rozhraní vedu vodiče out na IR+ a GND na IR-. Vše je tady galvanicky odděleno (pomocí optočlenů rozhraní nebo vnitřního relé klimatizace).

Aby bylo co nejméně práce v racku, připojil jsem + červená, - modrá a D bílá (kroucená se zelenou). Pokud bychom nenašli správnou bílou, je ji třeba před připojením rozhraní identifikovat. Bude na ni +5 V.



V rozvaděči je pak připojení co nejkratší, protože červená, modrá a bílá jsou napojeny na +, - a D linky iDům.