

## ISM 433 MHz TCVR (vysílač a přijímač) pro soutěže CANSAT

Vladimír Váňa  
vladvana@volny.cz

### 1. Úvod

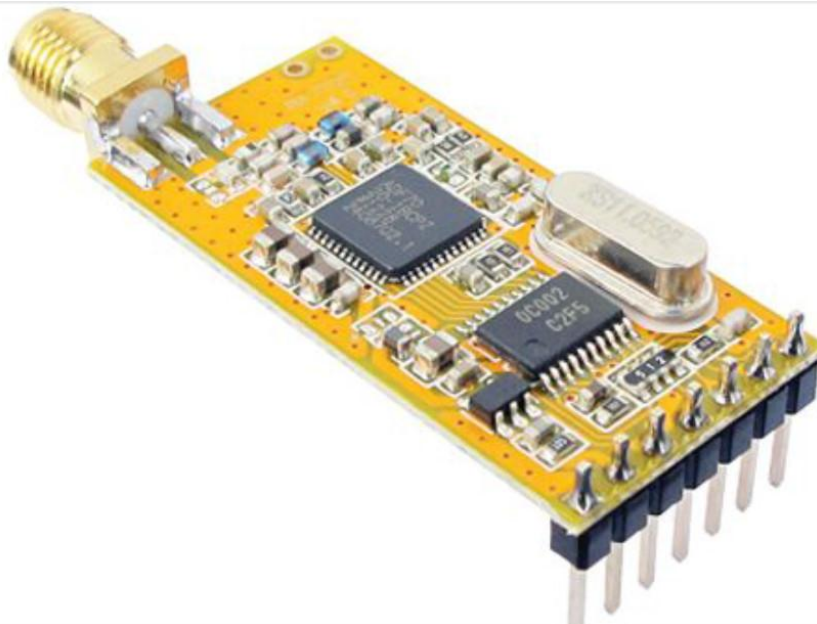
Ve třetím ročníku soutěže *European Cansat Competition 2013* byly používány trancievery APC220 a lze předpokládat, že i v soutěži v roce 2014 budou používány znovu jako součást startkitů od firmy T-minus. Proto jsem připravil tento materiál. Jeho 2 kapitola se může částečně krýt s obsahem materiálů, které získáme koncem ledna na workshopu učitelů v ESTEC.

Nikde jsem ovšem nenašel zapojení trancieveru ani firmware jeho mikrořadiče a tak jsem se pokusil v kap.3 o jeho analýzu. Je z ní zřejmé, že vůbec chybí anténní filtr a lze tedy předpokládat nevalné vlastnosti tcvr pokud jde o elektromagnetickou kompatibilitu (tj tcvr možná produkuje nekvalitní vf signál). Proto jsem asi také nenašel žádné certifikace ohledně elektromagnetické slučitelnosti udělené FCC či EU orgány.

Na druhé straně je pravda, že můžeme tcvr doplnit o vlastní anténní filtr a tím jeho vlastnosti zlepšit.

### 2. Základní informace o RF7020

Tranciever RF7020, jehož základem je obvod vysílače/přijímač ADF7020 firmy Analog Devices vyrábí pod názvem **APC220** čínská firma *SHENZHEN APPCON TECHNOLOGIES CO.LTD* ( [www.appcon.com.cn](http://www.appcon.com.cn) ) a pod názvem **DRF7020D20** firma *DORJI Applied Techmologies*.( [www.dorji.com](http://www.dorji.com) ) Tranciever se vyznačuje malými rozměry 37.5mm x 18.3mm x 7.0mm. Z jeho obrázku je zřejmé jeho konstrukční provedení:



#### Výrobci uvádějí následující parametry trancieveru:

Frekvenční rozsah 418MHz až 455MHz s krokem 1kHz, kanály po 200kHz

Výstupní výkon regulovatelný v 10 stupních , max. 20mW

GFSK modulace s modulační odchylkou 28,8kHz

UART interface s úrovní signálů TTL a rychlostí 1200,2400,4800,9600b,19200,38400,57600

Rychlost vysílání na vf 2400bps,4800bps,9600bps,19200bps

Napájecí napětí 3,4 až 5,5 V

Zapojení konektoru:

pin	jméno	funkce	
1	GND	Společná země	GND
2	VCC	napájení	napájení
3	EN	enable	vstup
4	RxD	Vstup UART	vstup
5	TxD	Výstup UART	výstup
6	AUX	Indikace data in/data out	výstup
7	SET	Nastavení, aktivní při L	vstup

### Nastavování a komunikace PC s TCVR

Obyčejně jsou dodávány dva transcievery (předpokládáme totiž oboustranné spojení) spolu s USB adapterem.



Vidíme ho i na obrázku:



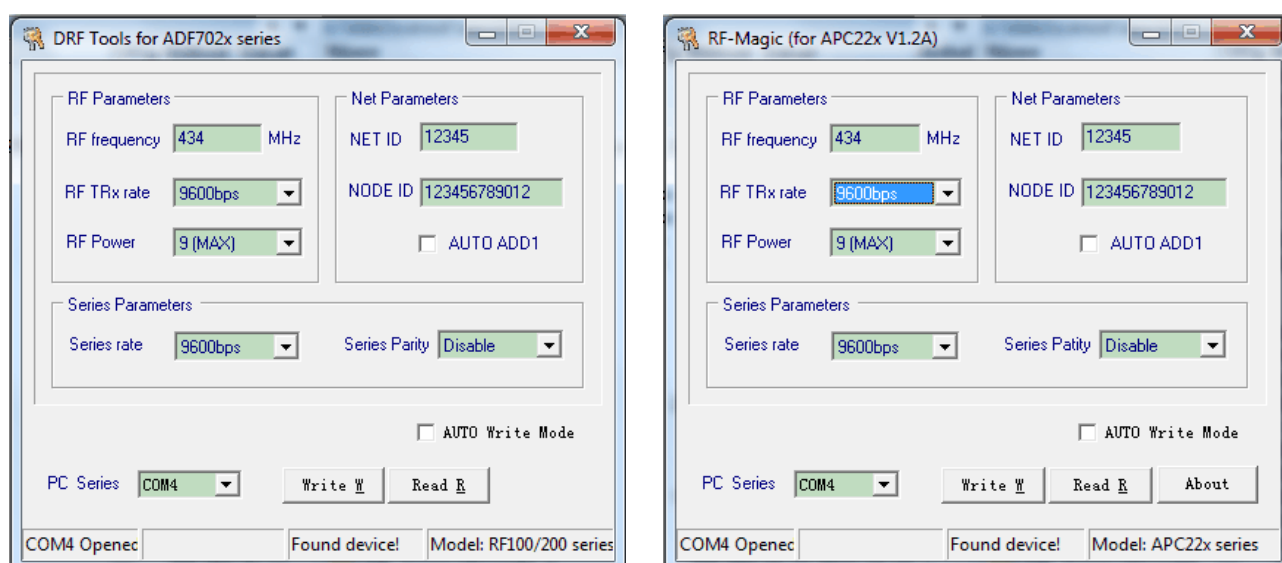
Je to převodník mezi USB a UARTem na úrovni TTL od firmy DFRobot.com. Obsahuje obvod CP2012 firmy Silicon Laboratories z jejichž stránek

<http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>

také stáhneme potřebné drivery.

Převodník má na jedné straně USB, na straně druhé signály TxD, RxD a RTS. Při jeho propojení s tcvr se signál TxD převodníku spojí s RxD trancieveru, RxD převodníku s TxD trancieveru a RTS převodníku s EN trancieveru. Dále jsou ještě propojeny GND a kladné napájecí napětí VCC. Piny 6 a 7 trancieveru nejsou s převodníkem propojeny. Pokud máme výše uvedený převodník propojen s tcvr, můžeme si přečíst konfiguraci tcvr či provést nové nastavení.

Pro nastavování TCVR můžeme použít program **RF-Magic (for APC22x v.1.2A)** nebo prakticky totožný **DRF Tools for ADF702x series**



Pokud je vše v pořádku, program nalezne tcvr (Found device) i jeho model. Musíme mít ovšem PC Series nastavený na virtuální sériový port odpovídající převodníku USB a rovněž přenosovou rychlost UARTu tcvr. Ta je defaultně 9600 Bd. Činnost programu je velice jednoduchá: Po kliknutí na tlačítko **Read** přečte program nastavení tcvr. Pokud naopak vyplníme textová políčka programu a klikneme na tlačítko **Write**, nakonfiguruje se těmito hodnotami tcvr.

Program mi nepracoval na 64 bitových WIN7. Na 32bitových WIN7 a na WIN XP byl funkční. V některých případech přestal pracovat a bylo nutné restartovat PC nebo alespoň ukončit tento program pomocí *Správce úloh Windows*. Především potřeboval výše uvedený převodník z USB s CP2012. Pokud jsem ale použil tento převodník jen se signály RxD a TxD, bez RTS, program nepoznal připojený tcvr. Odzkoušel jsem, že převodník s CP2102 lze sice použít jen se signály TxD a RxD, ale program rozpozná tcvr až poté, co pin 3 tcvr tj EN na chvíli připojíme na L a poté na chvíli na H. Od toho okamžiku je pak tcvr v konfiguračním režimu.

Vysvětlení a popis této vlastnosti tcvr jsem v dostupné dokumentaci nenašel. Tam se naopak tvrdí, že do konfiguračního režimu se tcvr dostane úrovní L na pinu 7 (SET). Ten však není k adapteru USB připojen a program ho tedy nevyužívá na rozdíl od programu *arduino* uvedeného v tomto textu o tři strany dále.

Rozchodit program s USB převodníkem s FT232RL se mi nepodařilo, byť po připojení EN k L a poté k H chvílkami v patičce programu publikoval nápis **Found device** a **Model:RF100/200 series ...**

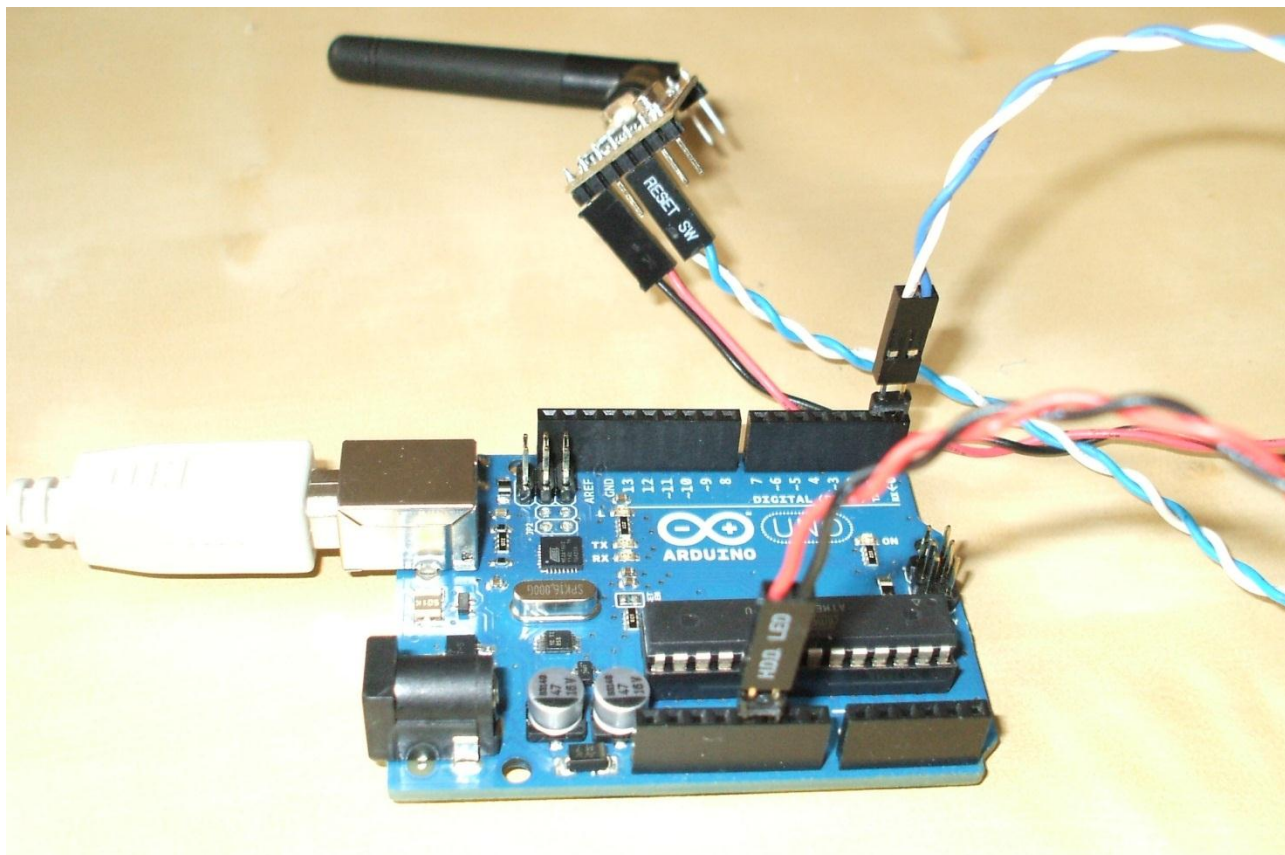


Dále provedeme ověření funkčnosti obou transceiverů. Využijeme přitom *Arduino Uno*, do něj uložíme následující kód:

```
int val = 0;
int ledPin = 13;
void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

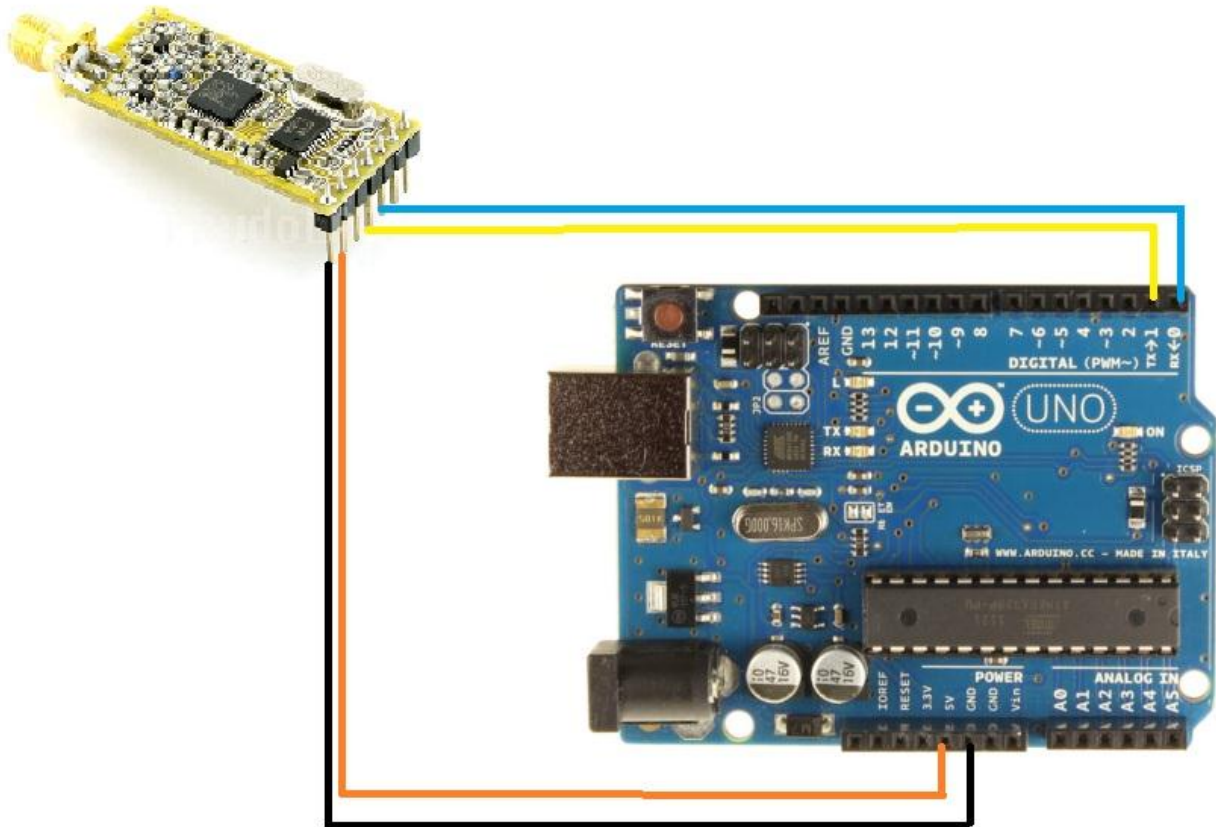
void loop()
{
  val = Serial.read();
  if (-1 != val) {
    if ('A' == val || 'a' == val) {
      Serial.println("Arduino for CANSAT CZ");
    } else if ('B' == val || 'b' == val) {
      digitalWrite(ledPin, HIGH);
      delay(500);
      digitalWrite(ledPin, LOW);
    }
  }
}
```

K tomuto *Arduino Uno* připojíme jeden z transceiverů:





Všimněme si, že k připojení stačily čtyři vodiče: červeným (+5V) a černým (GND) jsme připojili k TCVR napájení z *arduina*. To je napájeno 5V přes USB kabel z PC. Dále propojíme RxD a TxD TCVR s TxD a RxD *arduina* (překříženě):



Druhý TCVR připojíme přes redukci USB – UART (TTL) (třeba tu s CP2012 od *DFRobot.com*, která se dodává s APC220) k PC. Na PC spustíme nějaký terminálový program. Nastavíme u něj přenosovou rychlost 9600Bd, vysílání bez parity a to na virtuálním portu, k němuž přes USB máme připojen TCVR. Jestliže nyní vyšleme znak **b** nebo **B** je tento přijat druhým TCVR a odeslán do *arduina*. Na něm běžící kód jako reakci na příjem tohoto znaku provede na 500 ms bliknutí žlutou LED na desce *arduina* připojenou k pin 13.

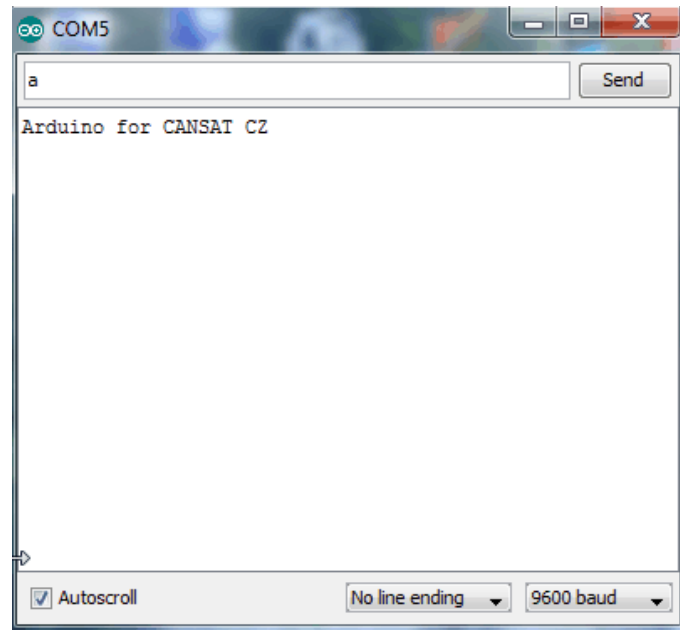
Další možností je odeslání znaku **a** nebo **A**. Po jeho příjmu na straně *arduina* a jeho vyhodnocení kódem na *arduinu* dojde k odeslání řetězce **Arduino for CANSAT CZ** prostřednictvím TCVR připojeného k *arduinu* a jeho příjmu druhým TCVR a následným zobrazením terminálovým programem na PC. Tím je ověřena funkčnost obou TCVR. Poznamenávám, že oba TCVR byly nastaveny naprosto stejně, tj. nejenom kmitočet, modulace, rychlost přenosu na vř straně i na straně UARTu 9600 Bd, ale i stejné (dokonce původní, defaultní) hodnoty NET ID a NODE ID.

# cansats in europe

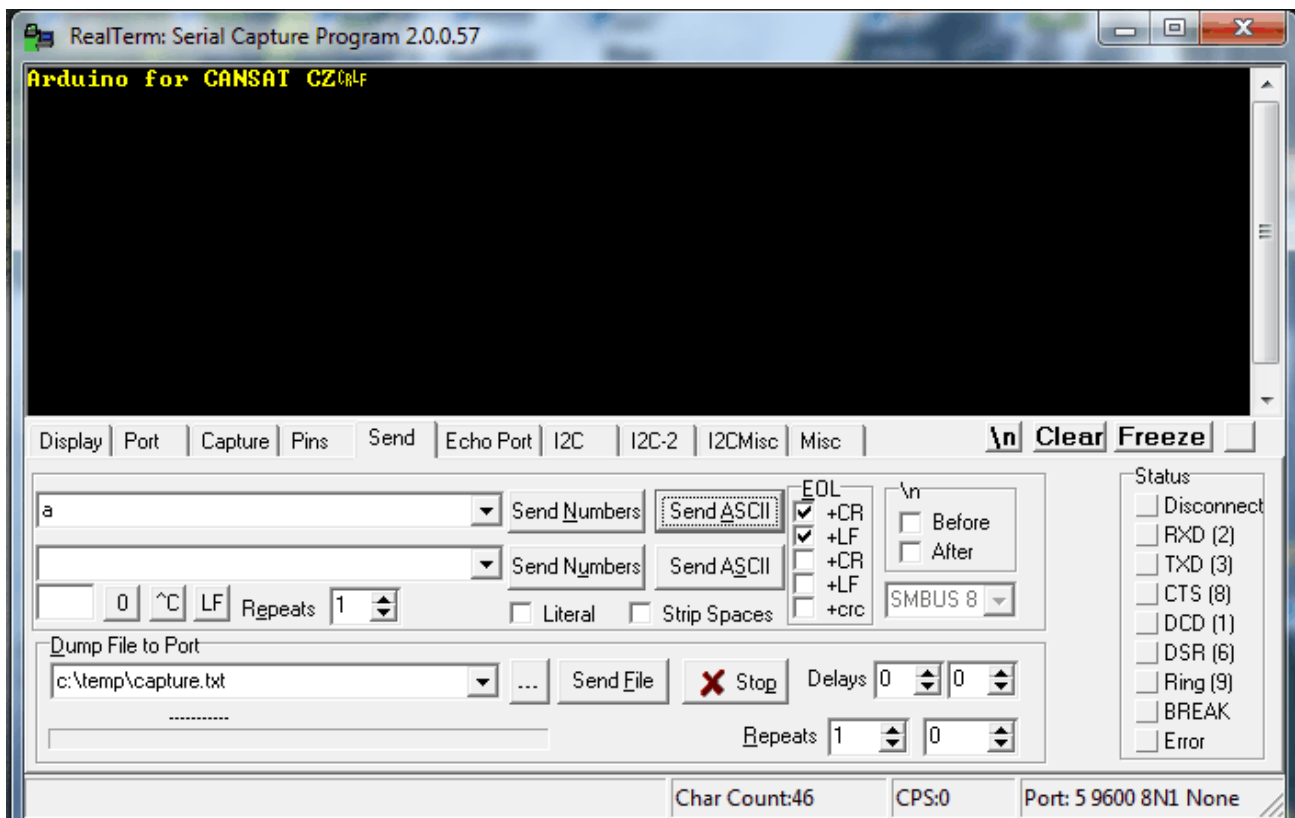
## 2014 european competition · pragsat team

---

Jako terminálový program jsem odzkoušel jednak terminálový program, který je součástí vývojového prostředí *arduino*:

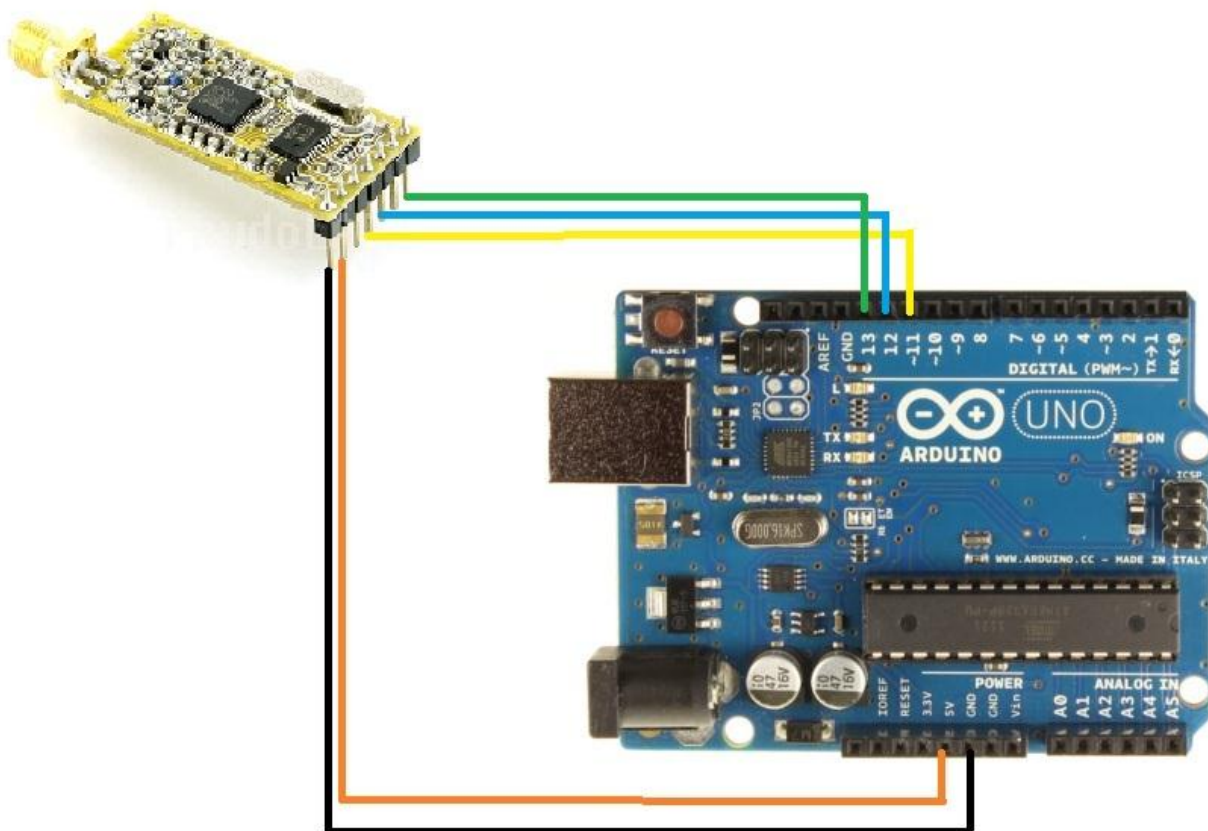


jednak RealTerm:



Vysílání obou TCVR jsem ještě sluchově sledoval příposlechem pomocí **TCVR Kenwood TM455** pro radioamatérské pásmo 70cm.

Nyní ukážu ještě jeden způsob konfigurace transceiverů APC220 využívajícího nastavení signálu SET (pin 7 tcvru) na L a následně přes RxD a TxD defaultní rychlostí 9600 Bd provádět příslušná nastavení. Využijeme k tomu *arduino*. Tentokrát ale připojíme TCVR k jeho pinům 11, 12 a 13:  
pinRX = 12 pinTX = 11 a pinSET= 13



Do *arduino* nahrajeme kód:



```
sketch_dec29b | Arduino 1.0.5
File Edit Sketch Tools Help
sketch_dec29b
#include <SoftwareSerial.h>
const int pinRX = 12;
const int pinTX = 11;
const int pinSET= 13;
SoftwareSerial tcvr(pinRX, pinTX);
// Crt softserial port and bind tx/rx to appropriate PINS
void setupSoftTCVR(void)
{
  pinMode(pinSET, HIGH);
  tcvr.begin(9600);
}

void setSettings(void)
{
  digitalWrite(pinSET, LOW); // pulling SET to low will put tcvr in
  delay(10); // stabilize tcvr.
  tcvr.println("WR 433900 3 9 3 0");
  delay(10); // ask for data
}

Done compiling.

Binary sketch size: 4 844 bytes (of a 32 256 byte maximum)

1 Arduino Uno on COM6
```

```
#include <SoftwareSerial.h>
const int pinRX = 12;
const int pinTX = 11;
const int pinSET= 13;
SoftwareSerial tcvr(pinRX, pinTX);
// Crt softserial port and bind tx/rx to appropriate PINS
void setupSoftTCVR(void)
{
  pinMode(pinSET, HIGH);
  tcvr.begin(9600);
}

void setSettings(void)
{
  digitalWrite(pinSET, LOW); // pulling SET to low will put tcvr in config mode
  delay(10); // stabilize tcvr.
```



```
tcvr.println("WR 433900 3 9 3 0");
delay(10); // ask for data
while (tcvr.available())
{
  Serial.write(tcvr.read());
}

digitalWrite(pinSET, HIGH); // put tcvr back in operation
delay(200);
}

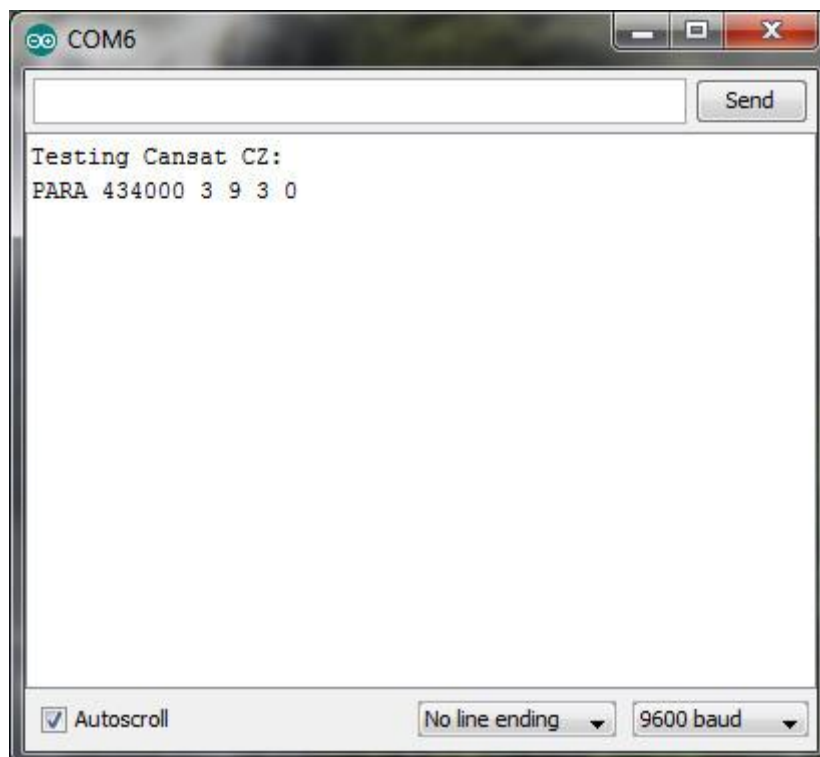
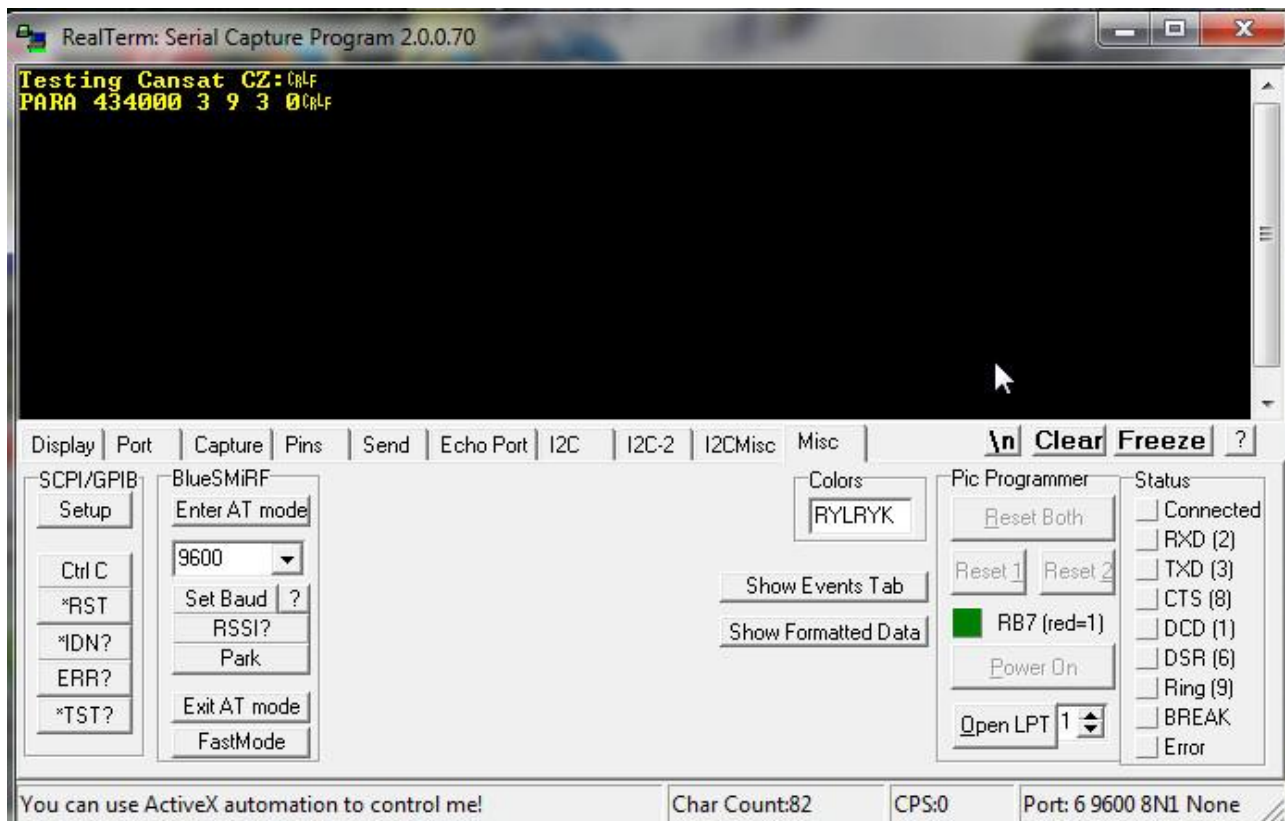
void getSettings(void)
{
digitalWrite(pinSET, LOW); // pulling SET to low will put tcvr in config mode
delay(10); // stabilize tcvr
tcvr.println("RD");
delay(10); // ask for data
while (tcvr.available())
{
  Serial.write(tcvr.read());
}
digitalWrite(pinSET, HIGH); // put tcvr back in operation
delay(200);
}

void setup()
{
Serial.begin(9600);
setupSoftTCVR();
Serial.println("Testing Cansat CZ:");
getSettings();
//setSettings();
}

void loop()
{
tcvr.println("Hello Cansat CZ!");
delay(5000);
}
```

Program pracuje tak, že jeho hlavní část proběhne hned na začátku, kdy se provádí funkce `setup()`. Ta nejprve nastaví přenosovou rychlost sériového kanálu (přes který komunikuje *arduino* s PC) na 9600 Bd. Na tuto rychlost nastaví rovněž softwarově vytvořený sériový kanál, k němuž je připojen TCVR. Nastavení přenosové rychlosti softwarového sériového kanálu se provede voláním funkce `setupSoftTCVR`. Následně se volá funkce `getSettings`. Ta nejprve nastaví na L úroveň pinu SET (7) TCVR. Poté odešle přes softwarový sériový kanál do TCVR řetězec znaků **RD** následovaný řídicími znaky `cr lf` (hexadecimálně 0x0D 0x0A). Protože při SET=L je TCVR v konfiguračním režimu, není tento řetězec odvílán přes vf, ale je chápán jako příkaz, aby odeslal zpět informace o svém nastavení.

To také `tcvr` provede ( v našem případě odeslal řetězec znaků **PARA 434000 3 9 3 0** následovaný řídicími znaky `cr lf`). Přijatý řetězec je poté odeslán přes druhý sériový kanál do PC, kde je zobrazen na terminálovém programu. Odkoušel jsem jak *RealTerm*, tak terminálový program obsažený ve vývojovém prostředí *arduino*:



Místo příkazu **RD** pro tcvr v konfiguračním režimu můžeme pro něj použít příkaz **WR** s příslušnými parametry (v našem případě **WR 433900 3 9 3 0**) a tak nastavit tcvr. Není tedy nutné používat k nastavení tcvr program **RF-Magic (for APC22x v.1.2A)** či **DRF Tools for ADF702x series**.

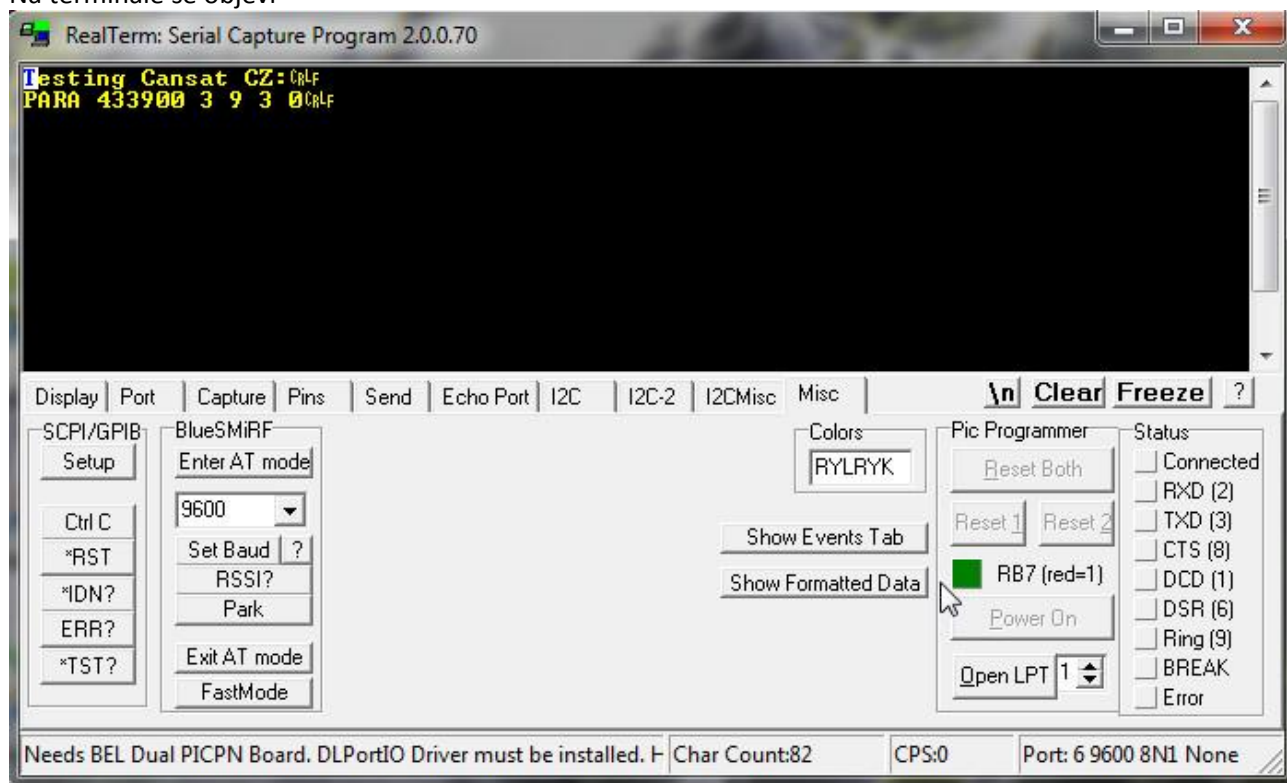
# cansats in europe

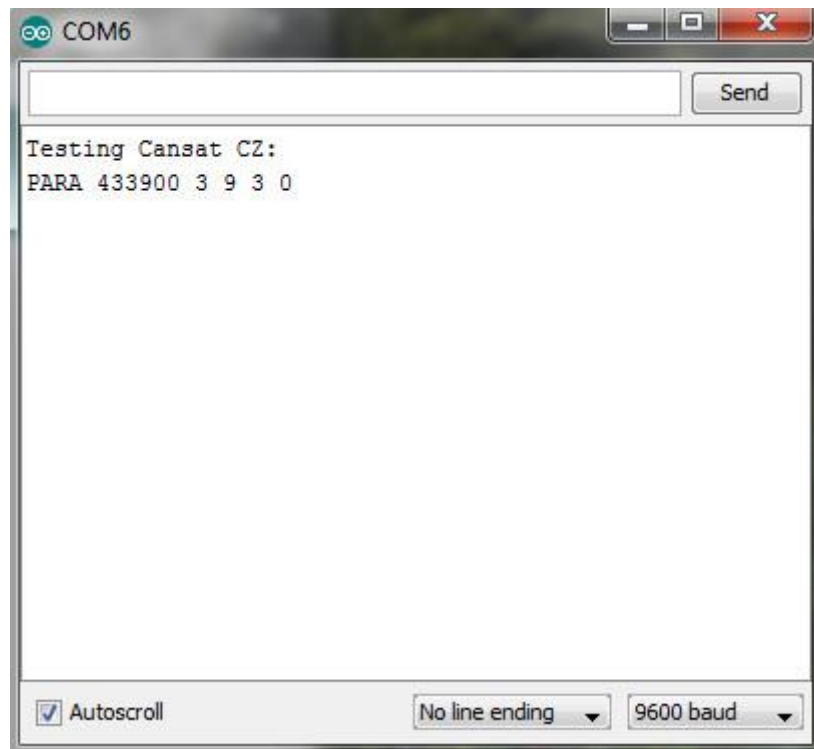
## 2014 european competition · pragsat team

Do *arduino* k tomuto účelu nahrajeme kód od výše uvedeného se lišícího jen ve funkci `setup()`:

```
void setup()
{
  Seriál.begin(9600);
  setupSoftTCVR();
  Seriál.println(„Testing Cansat CZ:“);
  //getSettings();
  setSettings();
}
```

Na terminále se objeví



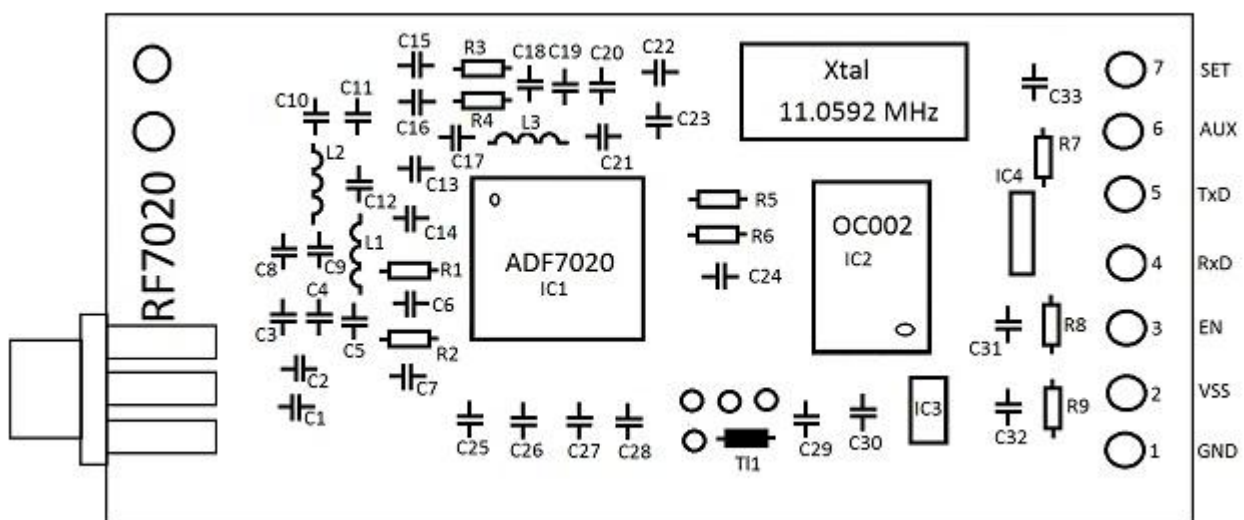
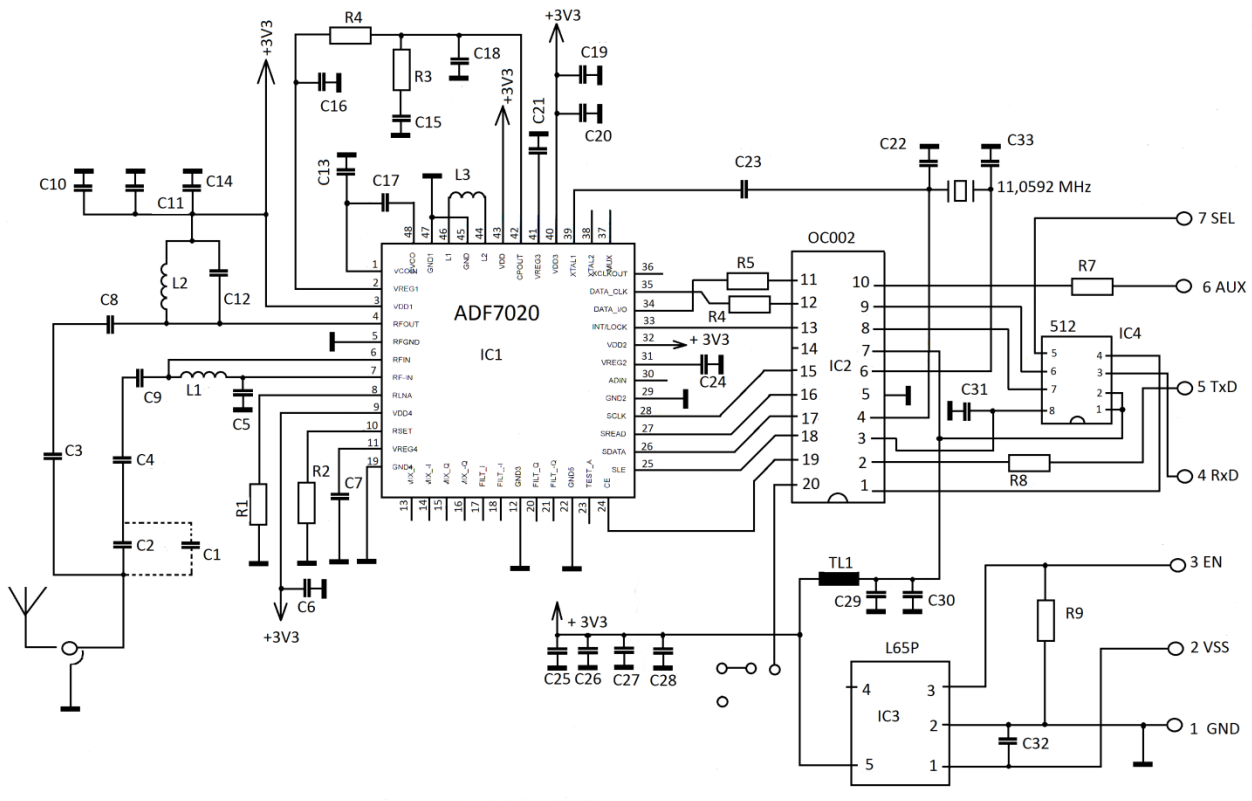


neboť se po změně konfigurace tcvr ještě provede její přečtení a následné odeslání na terminál. Ještě si vysvětlíme význam parametrů u WR a PARA. První parametr v **PARA 433900 3 9 3 0** je zřejmý. Je to kmitočet tcvr 433.9MHz. Druhý parametr je celé číslo 1, 2, 3 či 4 udávající přenosovou rychlost 2400, 4800, 9600 či 19200 Bd (b/s popř. bps) na vf (FSK). V našem případě máme nastaveno 9600 b/s. Třetí parametr je celé číslo 0, 1 až 9 udávající výstupní výkon. Nula odpovídá -1dBm, devítka pak maximální výkon 13dBm (20mW). My tedy máme nastavený maximální výstupní výkon. Čtvrtý parametr je celé číslo 0, 1, 2, 3, 4, 5 nebo 6 odpovídající rychlostem 1.2, 2.4, 4.8, 9.6, 19.2, 38.4 nebo 57.6 kb/s kterou se s tcvr komunikuje přes uart (piny RxD a TxD). I zde máme nastavenou rychlost 9600 Bd. Poslední parametr 0, 1 nebo 2 udává paritu. Nula znamená bez paritního bitu.

Pokud umíme pracovat s arduino, měla by nám tato kapitola 2 stačit k úspěšné práci na projektu CanSAT pokud jde o programovou obsluhu bezdrátového spojení v pásmu 433MHz mezi CanSATEm a pozemní stanicí.

### 3. Podrobnější informace o tcvr

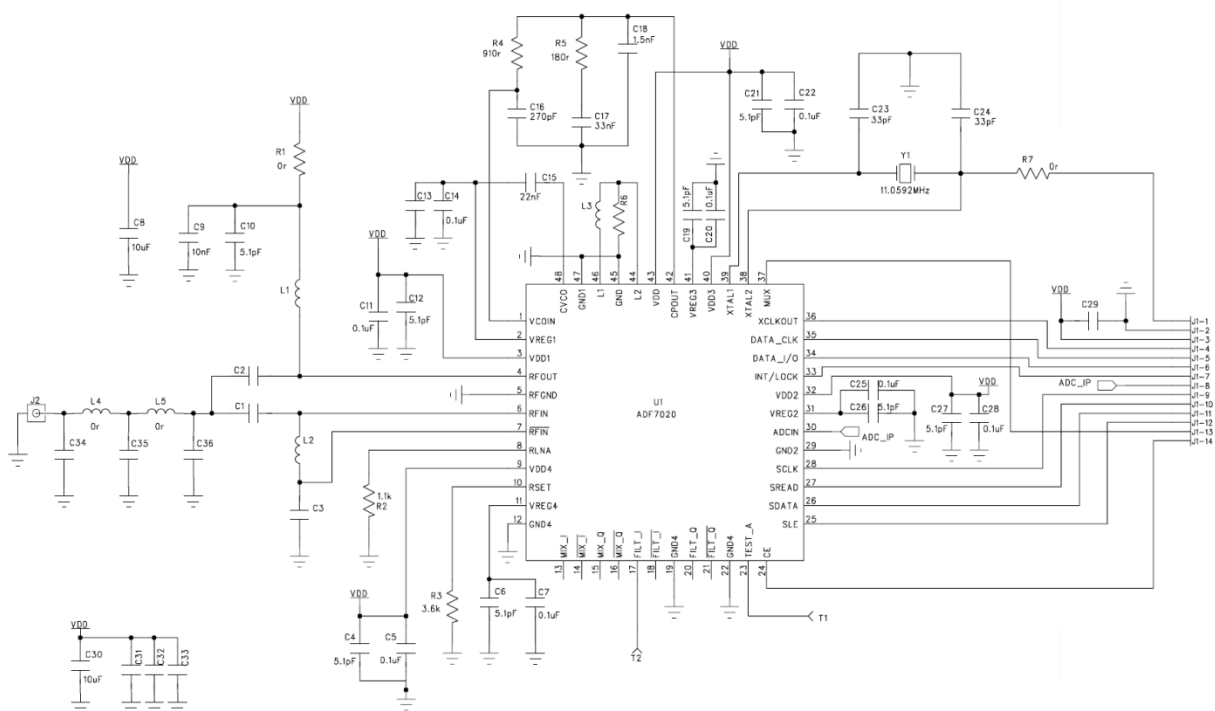
Vzhledem k tomu, že jsem neměl k dispozici podrobnější údaje o tcvr, nakreslil jsem na základě zapůjčeného exempláře transcieveru jeho schéma a rozložení součástí:

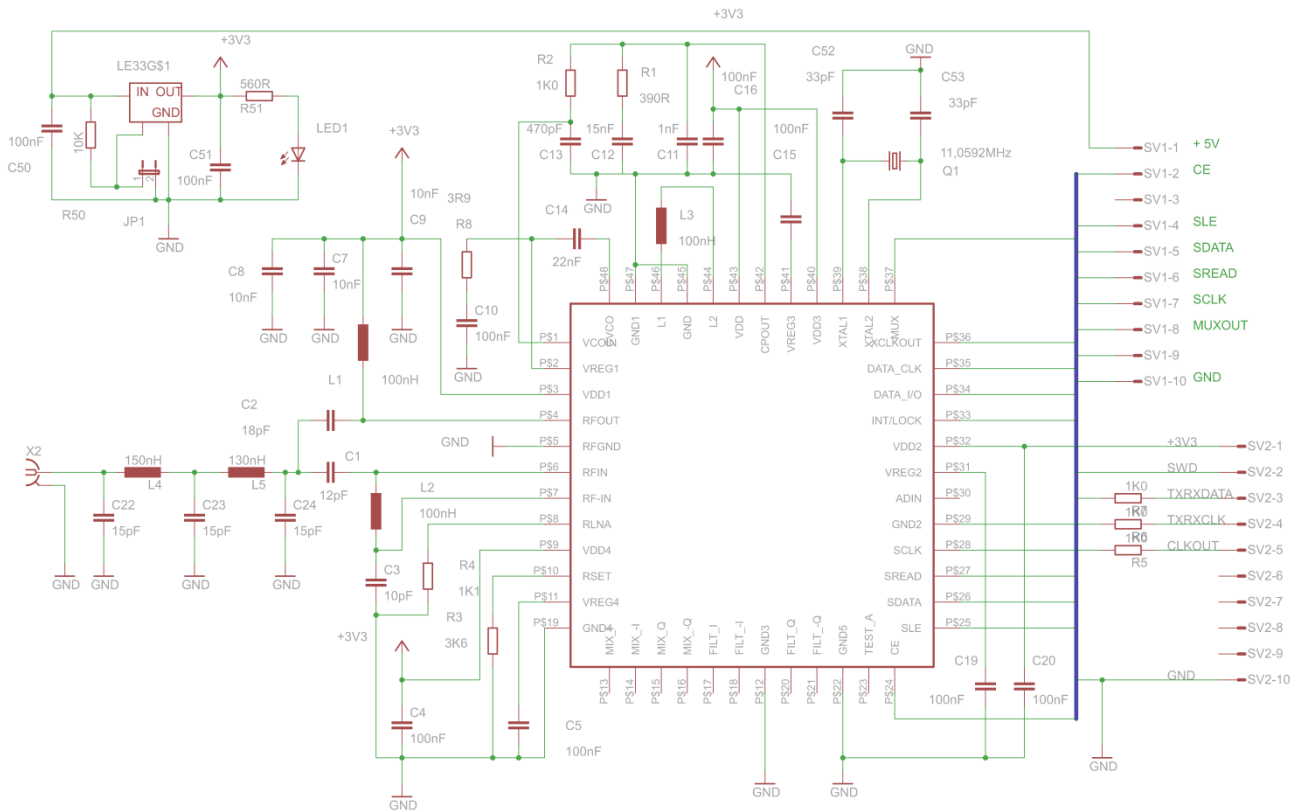


Tranciever se tedy skládá ze dvou částí – jednočipového trancieveru ADF7020 a mikrořadiče. Obvod ADF7020 je zapojen v doporučeném zapojení výrobce *Analog Devices*, viz. následující obrázek popř. obrázek další převzatý spolu s PCB deskou modulu vysílače z diplomové práce VUT Brno „FM vysílač APRS telemetrických dat v pásmu 144MHz“ Jana Bohátky z roku 2011. Autor diplomové práce uvádí, že modul © Vladimír Váňa 2014

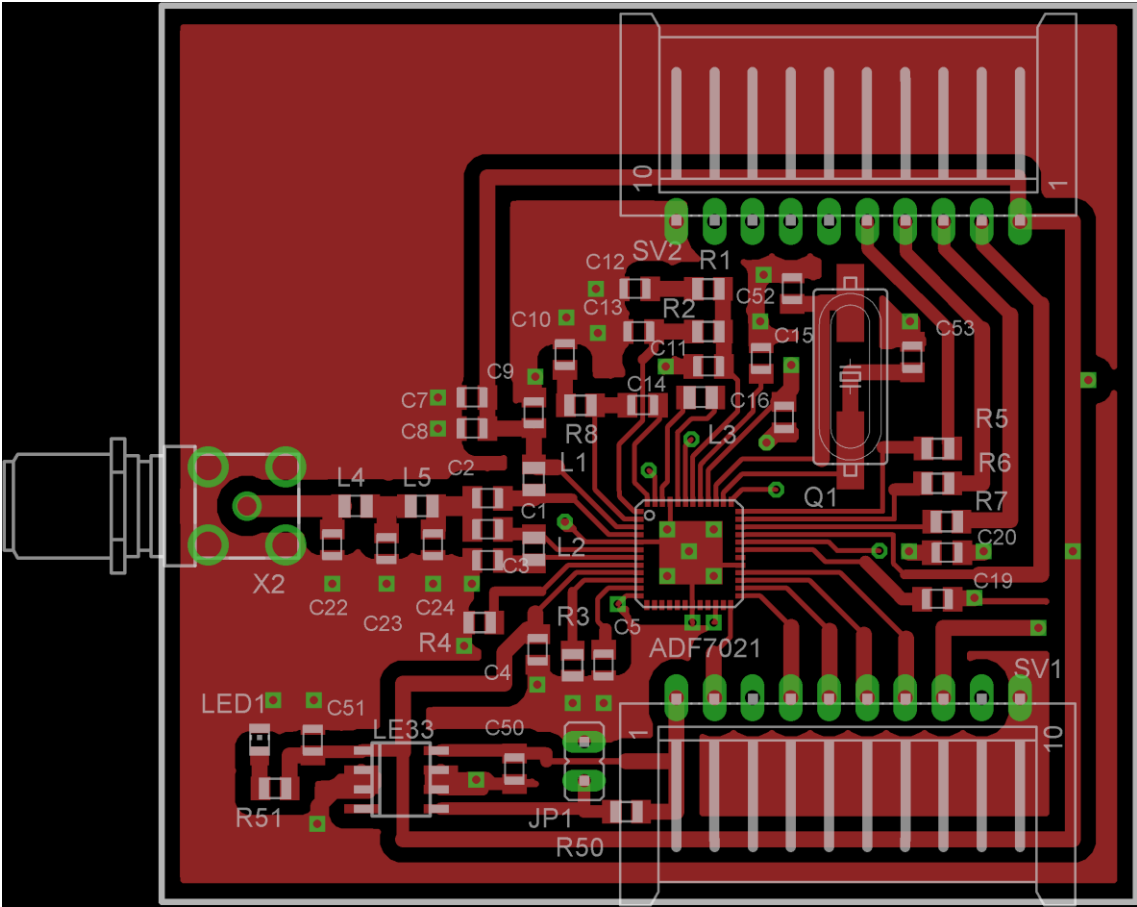


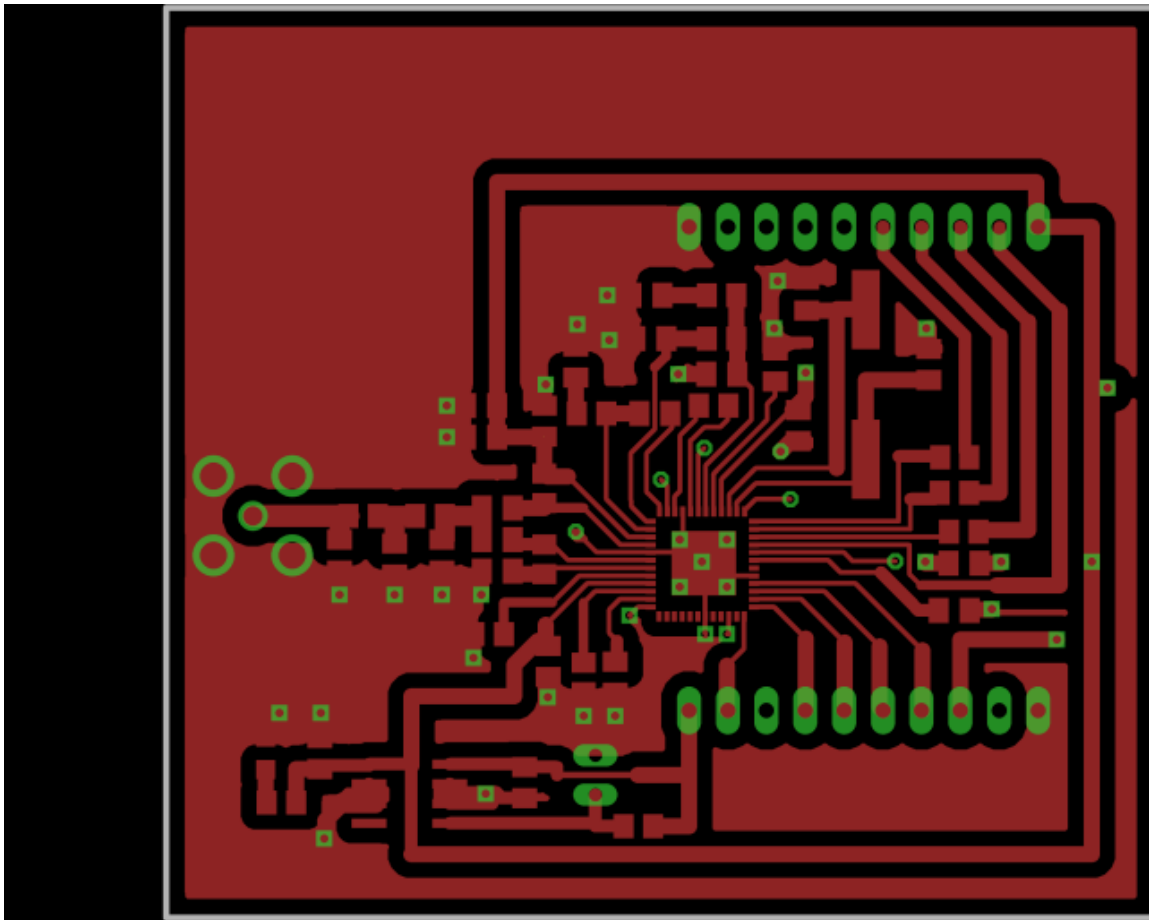
vysílače vychází z doporučeného zapojení od výrobce. Také rozmístění součástek bylo voleno jako je na vývojové desce od výrobce. Modul je osazený kompletně tak, že po dopsání ovládání by bylo možné pomoci tohoto modulu signál i přijímat. Ve své práci se totiž zabývá jen vysílací částí, tj. popisuje jen registry ADF7020 související s vysíláním, stejně jako kód firmware mikrořadiče ATmega164. *Moje pozn.:* Obvod ADF7020 či ADF7021 lze zdarma získat (2ks) jako tzv. **free samples** vyplněním žádosti na stránkách Analog Devices.

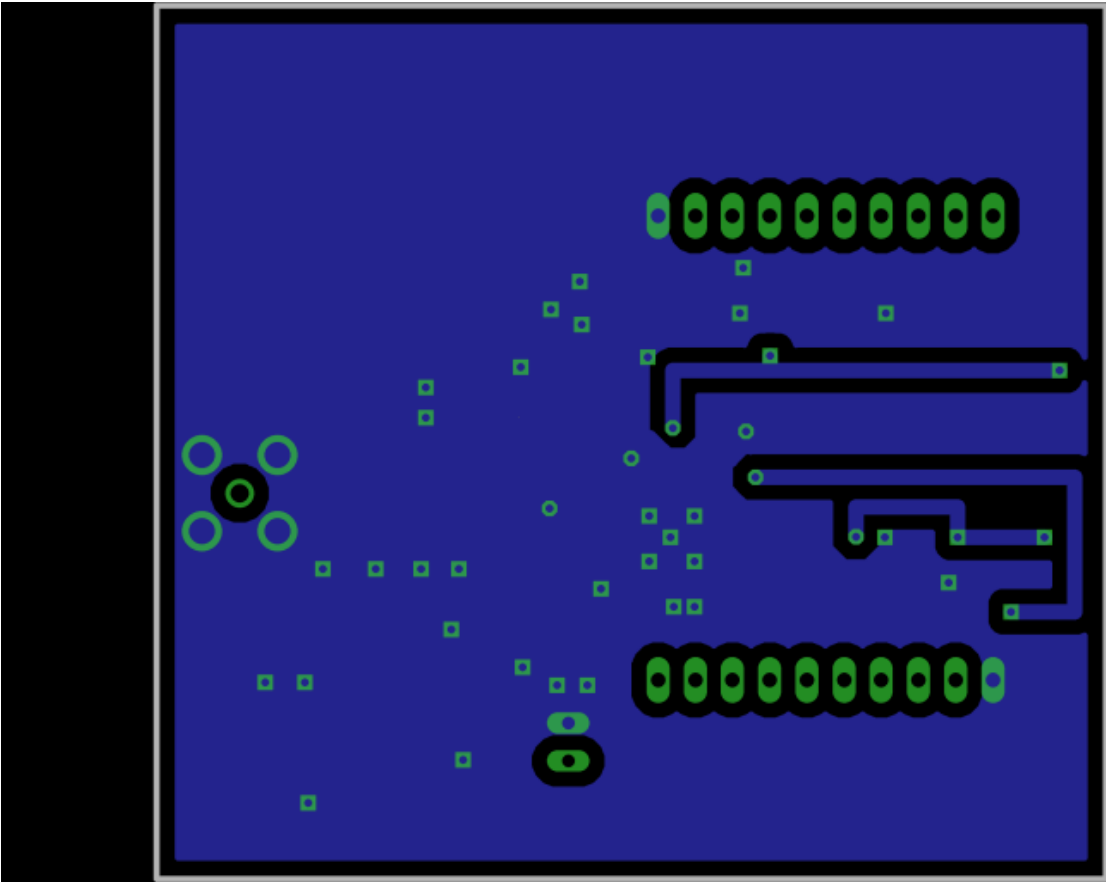




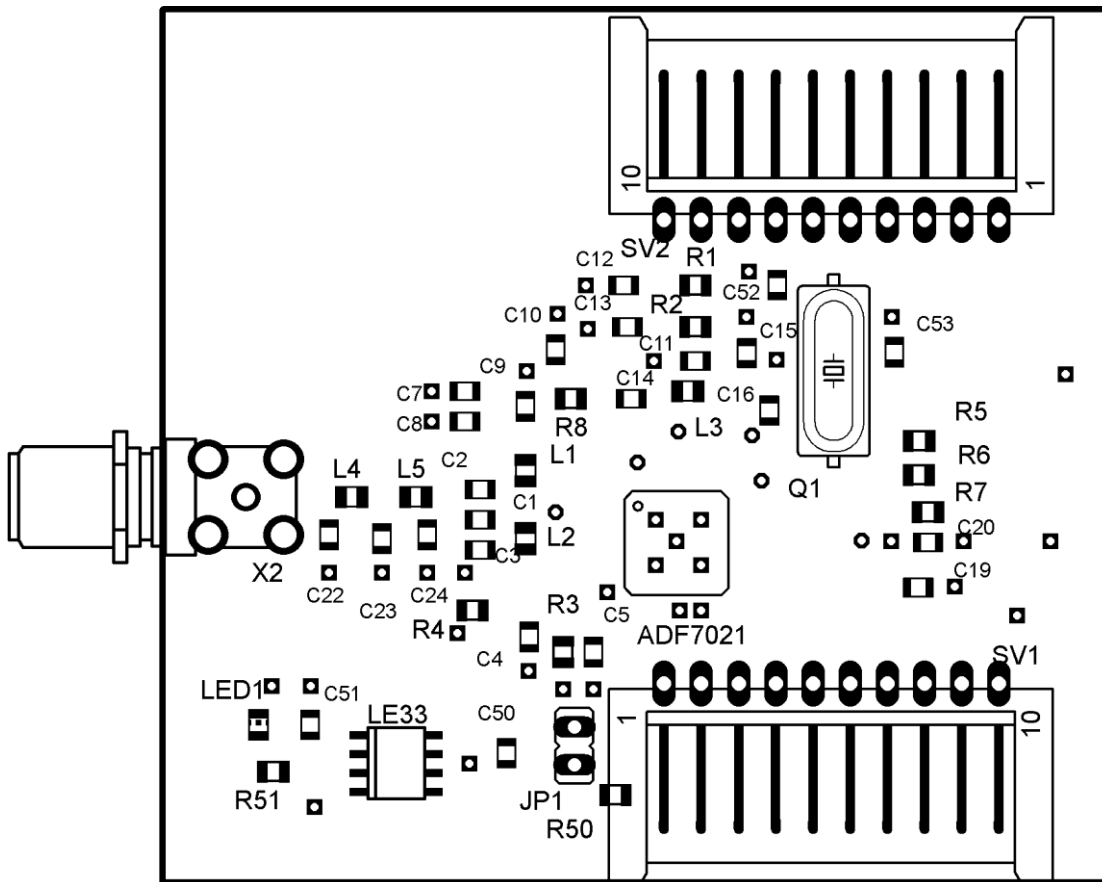
Rozměr desky PCB 96 x 65 [mm]:









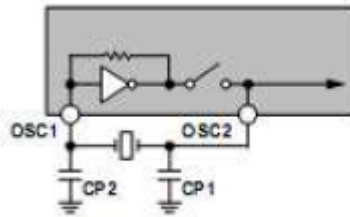


### Popis obvodu ADF7020 a ADF7021

V obvodu ADF7020 je zintegrován téměř celý přijímač a vysílač. Tento obvod umožňuje využívat modulace 2FSK, 3FSK, 4FSK na kmitočtech od 80 MHz do 650 MHz a od 862 MHz do 950 MHz. Na obrázcích výše uvedených je typické zapojení obvodu.

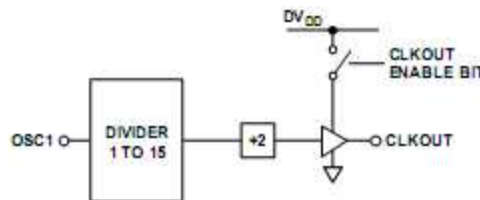
Vysílač obsahuje dva napětím řízené oscilátory (VCO) s výstupní chybou < 1 ppm. Protože obvod obsahuje 2 samostatná VCO je možné využít dvou-pásmový provoz, kde uživatel může na jedné frekvenci přijímat a na druhé frekvenci vysílat. V datasheetu jsou uvedeny tyto parametry, které platí pro VDD = 3 V a TA = 25 °C

Externí krystal by měl mít frekvenční toleranci < 10 ppm. Pro kmitání na správné frekvenci jsou potřeba dva paralelní rezonanční kondenzátory. Jejich hodnoty jsou závislé na parametrech krystalu a také na parazitních kapacitách na PCB. Typické hodnoty jsou v rozsahu 12 pF až 20 pF. Je výhodné použít kondenzátory s velmi nízkým teplotním koeficientem. Proud tekoucí do oscilátoru je možné měnit v rozsahu 20 uA až 35 uA pomocí nastavení bitu v registru R1\_DB(13:14).



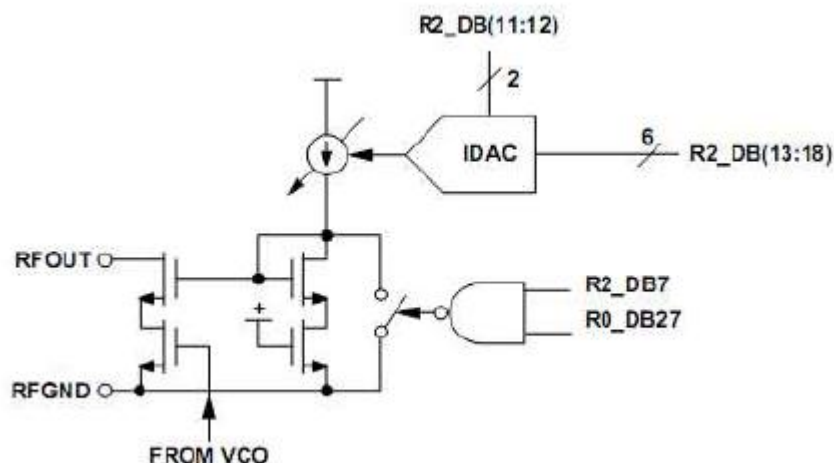
Obvod oscilátoru u ADF7020

Obvod ADF7020 obsahuje i referenční hodinový výstup. Na výstupu je obdélníkový signál se střídou 1:1. Frekvence signálu je dána frekvencí oscilátoru vydělenou děličkou. Z výroby je dělička nastavena na dělení 8. Hodnotu děličky lze měnit pomocí bitu v registru R1\_DB(7:10). Výstup CLKOUT lze použít jako kontrolu zda funguje komunikace mezi mikrokontrolérem a vysílačem, protože pro měření frekvence obdélníkového signálu není potřeba drahé přístroje.



Obvod CLKOUT

Výstup vysílače je obvod s otevřeným drainem (kolektorem), který byl navržený tak, aby do zátěže 50Ω na frekvenci 950 MHz dodal výkon + 13 dBm. Výstupní výkon se nastavuje pomocí bitu v registru R2\_DB(13:18).



Výstup ADF7020

Obvod obsahuje i obvody přijímače. Pokud by se signál z výstupu vysílače dostal na vstup přijímače, mohl by způsobovat chybný příjem a také poškodit vstupní obvody. Proto je vstup přijímače chráněn tak, že je v době vysílání obvodu zkratován.

Činnost obvodu ADF7020 je řízena obsahem jeho 16ti 32bitových registrů. Jejich význam najdeme v datasheetu. Jako příklad si nyní uvedeme popis čtyř registrů vysílací části ADF7020:

### Register 0 - N Register

Funkce	MUXOUT			UART MODE	Tx/Rx	8-Bit Integer_N								15-Bit Fraction_N								Address Bits										
bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
jméno	M3	M2	M1	U1	TR1	N8	N7	N6	N5	N4	N3	N2	N1	M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	C4	C3	C2	C1

Výchozí hodnota R0 je 0x10D0 5550

MUXOUT            0b000

UART Mode        0b1 → UART Mode Enable

Tx/Rx                0b0 → Transmit

8-Bit Integer N    0b0001 1010                    26D

15-Bit Fraction N 0b000 0101 0101 0101                    1365D

Address Bits        0b000

Příklad výpočtu

$$PFD = \frac{XTAL}{R\_Conter} = \frac{11,0592MHz}{1} = 11059200$$

$$RF_{OUT} = PFD * 0,5 * \left( Integer\_N + \frac{Fraction\_N}{2^{15}} \right)$$

$$RF_{OUT} = 11059200 * 0,5 * \left( 26 + \frac{1365}{2^{15}} \right) = 143,99MHz$$

Frekvenci lze měnit s krokem cca 170Hz



R-Cosine Alpha	0b0 → Raised Cosine Alpha 0,5 (Default)
TxDATA Invert	0b00 → TxData Invert Normal
Tx Frequency Deviaton	0b0 0001 0010      18D
Power Amplifer	0b00 0111 7D
PA Bias	0b10 → PA Bias 9uA
PA Ramo	0b000 → No Ramp
PA Enable	0b1 → Power Amplifier ON
Modul. Schema	0b100 → OverSamples 2FSK
Address Bits	0b0010

Příklad výpočtu

$$F_{DEV}[Hz] = 0,5 * \frac{PFD * Tx\_Frequency\_Deviation}{2^{16}} = 0,5 * \frac{11059200 * 18}{2^{16}} = 1518Hz$$

Výchozí hodnota R3 je 0x29BC 3913

### Register 3 - Transmit / Receiver Clock Register

Age Clk Divide						Sequencer Clk Divide						Cdr Clk Divide						Dem Clk Divide		Bbos Clk Divide		Address Bits									
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GD6	GD5	GD4	GD3	GD2	GD1	SK8	SK7	SK6	SK5	SK4	SK3	SK2	SK1	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	OK4	OK3	OK2	OK1	BK2	BK1	C4	C3	C2	C1

Age Clk Divide	0b00 0101	5D
Sequencer Clk Divide	0b0 0110 1111	111D
Cdr Clk Divide	0b00111001	57D
Dem Clk Divide	0b0001	1D
Address Bits	0b0011	



Příklad výpočtu

$$DataClk = \frac{XTAL}{Demod\_CLK\_Divide * Cdr\_Clk\_Divide * 32} = \frac{11059200}{1 * 57 * 32} = 6,063KHz$$

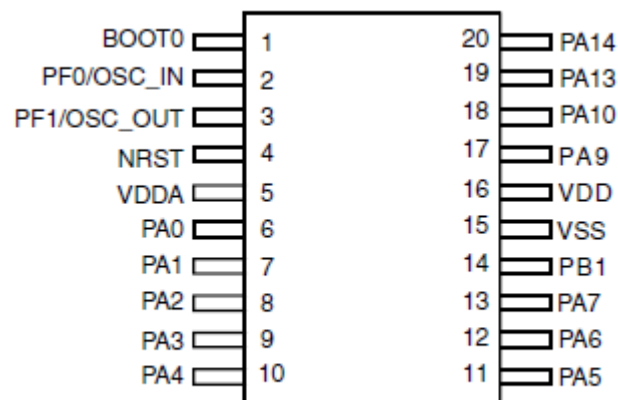
#### 4. Naměřené údaje

TO DO – až se dostanu k spektrálnímu analyzátoru

#### 5. Poznámky k uvažované vlastní konstrukci TCVR s ADF7020

V případě vlastního vývoje tcvr by jeho první částí mohla být výroba a osazení desky modulu vysílače dle diplomové práce nejlépe ve dvou provedeních a to pro pásmo 2m a pro pásmo 70cm. Další částí vývoje by byla na základě datasheetu ADF7020 tvorba firmware. Jako desku řízení bych pro účely vývoje předpokládal jednak arduino, jednak startkit STM32 s ARM Cortex od STMicroelectronic. Odladěný sw by se dal snadno portovat do STM32F050F6.

Jeho vlastnosti jsou pro naše účely více než dostačující, především pouzdro TSSOP20:



Nabízí se proto možnost tcvr vybavit tímto MCU spolu s ADF7020 a navrhnout PCB velikostí, rozložením součástí i obrazcem spojů velice podobným APC220. Další možností se jeví využití integrovaného sub GHz tcvr SPIRIT1 od STMicroelectronics (objednal jsem free samples na webu STMicroelectronics).

#### Příloha:

##### STM32F050xx

##### Features

- Core: ARM 32-bit Cortex™-M0 CPU, frequency up to 48 MHz
- Memories
  - 16 to 32 Kbytes of Flash memory
  - 4 Kbytes of SRAM with HW parity checking
- CRC calculation unit
- Reset and supply management

- Voltage range: 2.0 V to 3.6 V
- Power-on/Power-down reset (POR/PDR)
- Programmable voltage detector (PVD)
- Low power modes: Sleep, Stop and Standby
- VBAT supply for RTC and backup registers
- Clock management
- 4 to 32 MHz crystal oscillator
- 32 kHz oscillator for RTC with calibration
- Internal 8 MHz RC with x6 PLL option
- Internal 40 kHz RC oscillator
- Up to 39 fast I/Os
- All mappable on external interrupt vectors
- Up to 25 I/Os with 5 V tolerant capability
- 5-channel DMA controller
- 1 × 12-bit, 1.0 μs ADC (up to 10 channels)
- Conversion range: 0 to 3.6V
- Separate analog supply from 2.4 up to 3.6 V
- Up to 9 timers
- 1 x 16-bit 7-channel advanced-control timer for 6 channels PWM output, with deadtime generation and emergency stop
- 1 x 32-bit and 1 x 16-bit timer, with up to 4 IC/OC, usable for IR control decoding
- 1 x 16-bit timer, with 2 IC/OC, 1 OCN, deadtime generation and emergency stop
- 1 x 16-bit timer, with IC/OC and OCN, deadtime generation, emergency stop and modulator gate for IR control
- 1 x 16-bit timer with 1 IC/OC
- Independent and system watchdog timers
- SysTick timer: 24-bit downcounter
- Calendar RTC with alarm and periodic wakeup from Stop/Standby
- Communication interfaces
- 1 x I2C interface; supporting Fast Mode Plus (1 Mbit/s) with 20 mA current sink, SMBus/PMBus, and wakeup from STOP
- 1 x USART supporting master synchronous SPI and modem control; one with ISO7816 interface, LIN, IrDA capability auto baud rate detection and wakeup feature
- 1 x SPI (18 Mbit/s) with 4 to 16 programmable bit frames, with I2S interface multiplexed
- Serial wire debug (SWD)
- 96-bit unique ID
- Extended temperature range: -40 to +105°C

### **SPIRIT1**

Low data rate, low power sub-1GHZ transceiver.

#### **Features**

- Frequency bands: 150-174 MHz, 300-348 MHz, 387-470 MHz, 779-956 MHz
- Modulation schemes: 2-FSK, GFSK, MSK, GMSK, OOK, and ASK
- Air data rate from 1 to 500 kbps
- Very low power consumption (9 mA RX and 21 mA TX at +11 dBm)
- Programmable RX digital filter from 1 kHz to 800 kHz
- Programmable channel spacing (12.5 kHz min.)
- Excellent performance of receiver sensitivity (-118 dBm), selectivity, and blocking
- Programmable output power up to +16 dBm
- Fast startup and frequency synthesizer settling time (6  $\mu$ s)
- Frequency offset compensation
- Integrated temperature sensor
- Battery indicator and low battery detector
- RX and TX FIFO buffer (96 bytes each)
- Configurability via SPI interface
- Automatic acknowledgment, retransmission, and timeout protocol engine
- AES 128-bit encryption co-processor
- Antenna diversity algorithm
- Fully integrated ultra low power RC oscillator
- Wake-up on internal timer and wake-up on external event
- Flexible packet length with dynamic payload length
- Sync word detection
- Address check
- Automatic CRC handling
- FEC with interleaving
- Digital RSSI output
- Programmable carrier sense (CS) indicator
- Automatic clear channel assessment (CCA) before transmitting (for listen-before-talk systems). Embedded CSMA/CA protocol
- Programmable preamble quality indicator (PQI)
- Link quality indication (LQI)

- Whitening and de-whitening of data
- Wireless M-BUS, EN 300 220, FCC CFR47 15 (15.205, 15.209, 15.231, 15.247, 15.249), and ARIB STD T-67, T93, T-108 compliant
- QFN20 4x4 mm RoHS package
- Operating temperature range from -40 °C to 85 °C

### Applications

- AMR (automatic meter reading)
- Home and building automation
- WSN (wireless sensors network)
- Industrial monitoring and control
- Wireless fire and security alarm systems
- Point-to-point wireless link

Figure 2. Suggested application diagram

