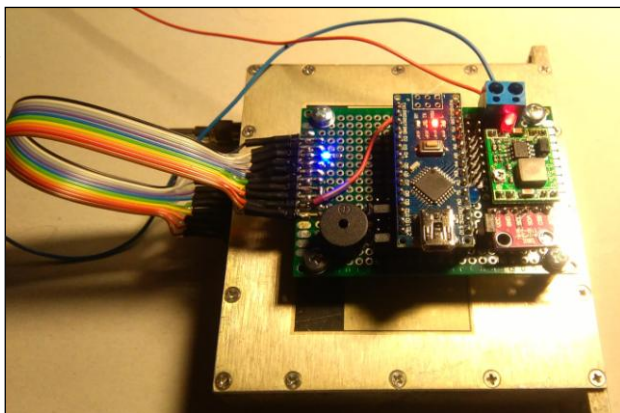


Arduino 10GHz weather beacon

Tomáš Krejča, OK1DXD, ok1dxd@seznam.cz

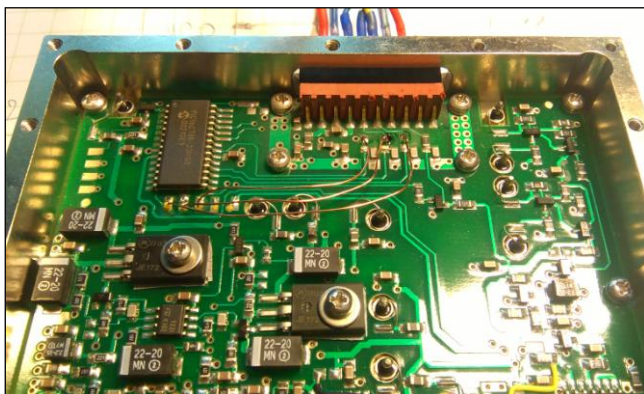
Mít vlastní „osobní“ maják má na mikrovlnných pásmech mnoho výhod, kromě sledování podmínek šíření jej často využijeme pro pohotové ověření funkčnosti a nastavení zařízení a antén. Srdcem majáku je oblíbený syntezátor DFS-1101 od firmy Elcom (Ceragon), který lze za velmi přístupnou cenu pořídit na eBay. Syntezátor poskytuje kvalitní a díky internímu OCXO i celkem stabilní signál.

Nevýhodou je základní krok 3,33 MHz, který je ale daný interním PIC, který přes SPI programuje interní PLL SIEMENS PMB2306 a nelze jej změnit. Proto jsem se rozhodl upravit jednotku syntezátoru, aby bylo možné přímo ovládat vnitřní PLL a odstranit tak zmíněné omezení.



Úprava DFS-1101

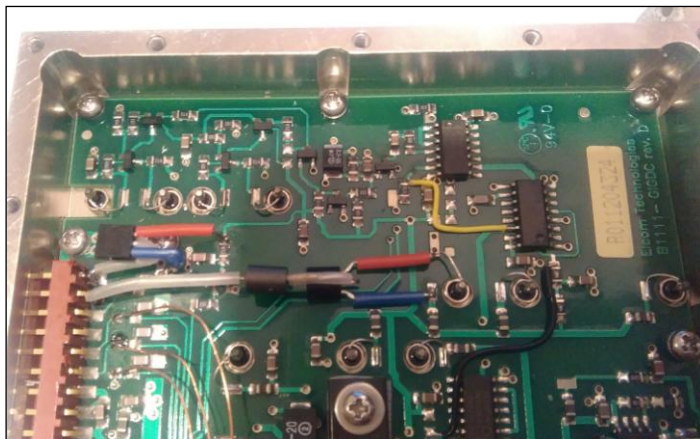
Jedná se o jednoduchou a rychlou úpravu, ale je třeba si předem dobře rozcvičit ruku, nebo použít akumulátorový šroubovák, jinak vás bude po povolení více jak 50ti šroubů bolet ruka. Začneme tím, že odpojíme PIC od řízení PLL a PLL propojíme přímo na SPI piny konektoru syntezátoru. Pro lepší orientaci je zde k dispozici detailní pohled.



Nyní bude nutné odstranit 3 ks odporů 10K na SPI sběrnici u konektoru. Tím odpojíte PIC od konektoru a zde pak naletujete nové vodiče (tenký Cu drát) mezi PLL a konektor. Dále odstraníme tři SPI propojovací vodiče mezi deskou PIC a druhou deskou na které je PLL. Uvolněnými otvory protáhneme nové vodiče a zaletujeme je u PLL. Tím je hotové nové propojení SPI rozhraní od PLL na výstupní konektor.

Na poslední pin konektoru připojit jednodrátové teplotní čidlo Dallas DS18B20, které je napájeno z desky plošného spoje v místě kde je vedení 5V viz obr. PullUp odpor mezi datovým vývodem a + 5V je zapojen na desce Arduino.

Pro zlepšení teplotní stability doporučuji nad OCXO vložit destičku z pěnové hmoty (některé DFS 1101 tuto teplotní izolaci mají již od výrobce, ale některé asi starší nemají).



Klíčování majáku

Je provedeno spínáním napájecího napětí násobiče. Tento stupeň je vybrán záměrně, aby bylo potlačené nosné frekvence v nezaklíčovaném stavu

co nejvyšší (klíčování v zesilovacím stupni mělo malou izolaci). Pokud je napájení odpojeno, tak na výstup prochází signál nosné asi jen -30dBm, pokud by to bylo na závadu, tak lze jednoduše stejným tranzistorem klíčovat i následující stupeň, nebo případný PA. Předposlední pin na konektoru syntezátoru použijeme pro přívod klíčování - je potřeba jej odstříhnout těsně nad plošným spojem - je totiž připojen na zem (GND). Pak připojíme odpor 10K mezi tento pin a bázi klíčovacího PNP tranzistoru (v mém případě jsem použil BC556A) Nezapomeneme na přívod do báze navléknout feritový korálek pro blokování pronikání VF signálu. Odpojíme původní drátovou propojku mezi deskou plošného spoje a průchodkou v krabíčce. Kolektor tranzistoru naletujeme na kontakt plošného spoje, kam bylo původně přivedeno napájecí napětí násobiče. Emitor pak na průchodku. Více je zřejmé z obrázku.

Arduino

V prototypu jsem použil Arduino UNO, ve finální verzi pak kvůli velikosti Arduino Nano, které je přes konektor připojeno na univerzální plošný spoj. Zde je také step down zdroj 6V pro syntezátor a piezo beep. Senzor, který měří venkovní teplotu, tlak a vlhkost doporučuji připojit na krátký kabel a umístit mimo krabici majáku, aby teplota uvnitř neovlivňovala naměřené hodnoty. Na desce je také na analogovém vstupu zapojen jednoduchý napěťový dělič, který upravuje hodnotu napájecího napětí (cca 12 V) na max. 5 V, které je schopno Arduino svým AD převodníkem ještě zpracovat. Protože byl záměr napájet maják na odlehých místech pomocí solárního článku, tak informace o úrovni napájecího napětí je též součástí měření a „telemetrie“ kterou vysílá maják.

Popis čidla BME280

BME280 je další generace miniaturního senzoru od firmy Bosch, která nahrazuje předchozí generaci BMP085/BMP180/BMP183. Pozor, k dispozici jsou dvě verze: verze BME280 (E=Environmental) umí na rozdíl od BMP280 kromě teploty a tlaku také měřit vlhkost okolního vzduchu. Protože maják byl určen pro sledování podmínek troposferického šíření radiových vln, tak doporučuji pořídit o trochu dražší verzi BME i s měřením vlhkosti.

Zajímavé je, že senzor umí komunikovat po SPI, nebo I2C rozhraní. Vzhledem k tomu, že SPI je již v majáku využité pro programování PLL rozhodl jsem se pro senzor použít I2C. Program očekává I2C ID adresu 0x77, pokud má vaše čidlo jiné ID (zpravidla 0x76, pak je

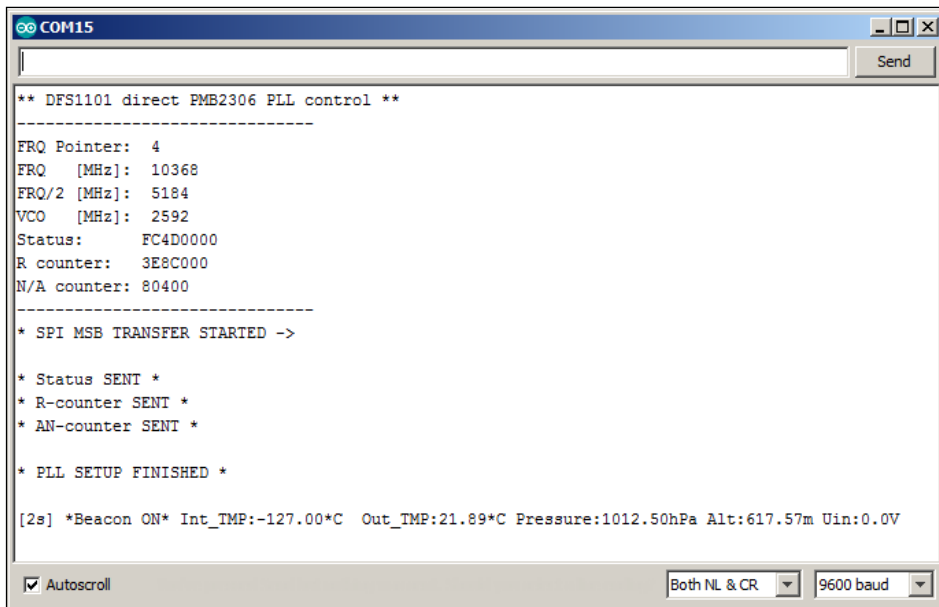
nutné tuto adresu změnit v knihovně Adafruit, nebo lze toho také obvykle dosáhnout pomocí adresovacího jumperu na desce čidla (na vstupu SDO = HIGH je pro ID 0x76, SDO = LOW je pro 0x77). Teplota je vypočtena ve stupních Celsia, pokud potřebujete Farenheita, je jednoduché to převést pomocí klasického vzorečku $F = C * 9/5 + 32$.

Popis ovládacího SW

Pro kompilaci budete potřebovat externí knihovny (například morse coder dle modifikace od AF4JF), jejich seznam je jako obvykle uveden na začátku Arduino sketche. Podle typu senzoru vyberte odpovídající knihovnu od firmy Adafruit. Vnitřní teplota syntezátoru je měřena čidlem Dallas DS18B20. Každé čidlo má vlastní unikátní ID, které je potřeba zjistit a dosadit do programu. Lze pak zapojit více těchto čidel na společný datový vodič a měřit např. napětí zdroje slunečních článků apod.

Co se týče nastavení PLL, tak hodnoty registrů Status, R, A/N čítače jsou pro předem zvolené kmitočty vypočteny externě v MS Excel tabulce (dík zde patří Pavlovi, OK1DX, který ji sestavil a prakticky ověřil). Arduino pak toto nastavení posílá pod SPI ve formě „telegramů“ odpovídající délky. Pokud je PLL správně naprogramovaná tak LED dioda, která indikuje zavěšení smyčky po zapnutí krátce blikne a pak už zůstane stabilně svítit. Koho to více zajímá, tak podrobnosti najde v komentářích v odpovídající sekci sketche. Na desce Arduino jsou dva jumpery, kterými lze přepínat mezi čtyřmi předem připravenými frekvencemi (nové nastavení se načte vždy při resetu Arduina). Další jumper slouží k přepnutí módu majáku, tedy mezi CW klíčováním syntezátoru a nebo jen morse výstupem na piezo bzučák pro akustickou kontrolu telemetrických údajů. Konkrétní zapojení pinů Arduina a syntezátoru viz tabulka.

Program běží ve smyčce, nejprve načte hodnoty ze senzorů a pak je postupně vysílá CW a současně posílá po seriovém rozhraní na SerialMonitor, takže během ožívání můžete sledovat hodnoty nastavení registrů PLL a vysílaný text s telemetrií.



```
COM15
Send

** DFS1101 direct PMB2306 PLL control **
-----
FRQ Pointer: 4
FRQ [MHz]: 10368
FRQ/2 [MHz]: 5184
VCO [MHz]: 2592
Status: FC4D0000
R counter: 3E8C000
N/A counter: 80400
-----
* SPI MSB TRANSFER STARTED ->

* Status SENT *
* R-counter SENT *
* AN-counter SENT *

* PLL SETUP FINISHED *

[2s] *Beacon ON* Int_TMP:-127.00*C Out_TMP:21.89*C Pressure:1012.50hPa Alt:617.57m Uin:0.0V

Autoscroll Both NL & CR 9600 baud
```

DFS1101 Beacon pin usage				
Arduino Nano				
digital D			DFS1101 pinout	
D0	system		1	6V
D1	system		2	12V
D2	TMP DS18B2 sensor	inside temperature	3	GND
D3	KEY	Key on LOW level	4	GND
D4	LOCK	input from DFS1101	5	SPI DATA
D5	SPI CE	SPI data to PLL	6	SPI CLK
D6	SPI CLK	SPI data to PLL	7	SPI EN/CE
D7	SPI DATA	SPI data to PLL	8	PLL LOCK
D8	HUM sensor	reserved	9	KEY
D9	Beep	must be a PWM pin!	10	TEMP
D10	FRQ1	Freq setup jumper		
D11	FRQ2	Freq setup jumper		
D12	Beacon mode switch			
D13	status LED	Arduino		
Analog A				
A0				
A1	CW KEY	LED		
A2				NANO pin us
A3	PLL LOCK LED	blue LED		
A4	I2C	SDA		
A5	I2C	SCL		
A6				
A7	Vin	Vin voltage divider		5K6/22K resistors
A8				

Závěr

Cílem bylo vytvořit jednoduchý maják pro sledování podmínek šíření. Jako postranní efekt získáme možnost přímo řídit PLL obvod v syntezátoru DFS1101 a rozšířit jeho využití nejen jako LO, ale i jednoduchého signálního generátoru v pásmu 10.300 až 11.400 MHz.

Užitečné informace:

- datasheet PLL
- datasheet Dallas
- datasheet
-
- soundfile
- OK1DX table
- Arduino sketch