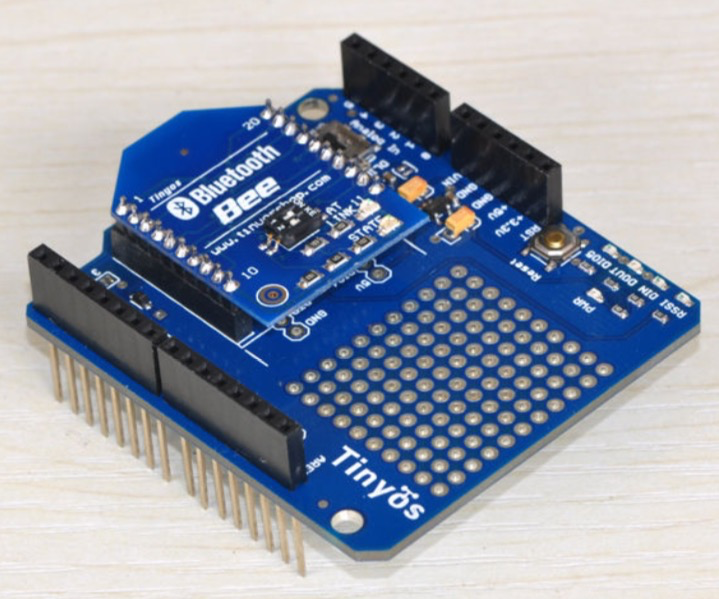
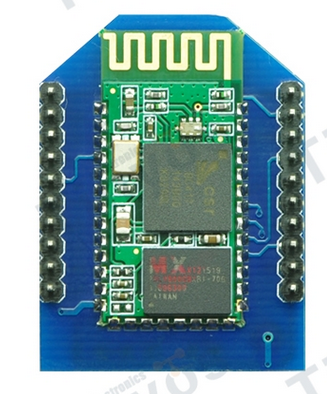
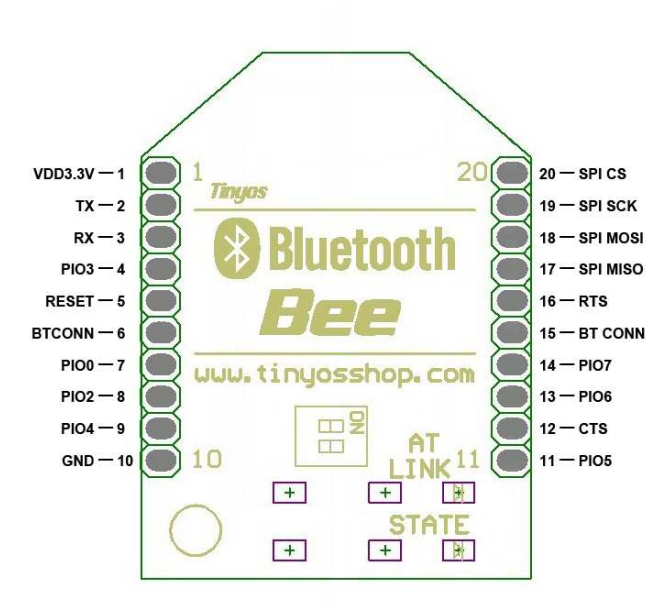
# TinyOS Bluetooth Bee & TinySine Bee Shield V1r1



## TinyOs bluetooth bee

Bluetooth Bee [link] je modul kompatibilný s platformou XBee [link] osadený obľúbeným ekonomickým bluetooth modulom HC-06 [obr]. XBee je registrovaná značka spoločnosti Digi International a označuje rodinu kompaktných modulov určených pre bezdrôtovú komunikáciu. Moduly sú charakterizované dvomi 10 pinovými kolíkovými lištami RM2 po stranách modulu. Pre komunikáciu s nadradeným mikrokontrolérom (MCU) postačia štyri vodiče: napájanie 3.3V, zem, komunikačná linka pre príjem dát RX (z MCU do XBee) a linka pre vysielanie dát TX (z XBee do MCU). Tieto piny majú fixne definovanú polohu [obr], čo umožňuje bezproblémové zamieňanie komunikačných modulov, či už sa jedná o Bluetooth, Wifi, Zigbee, alebo inú bezdrôtovú technológiu.





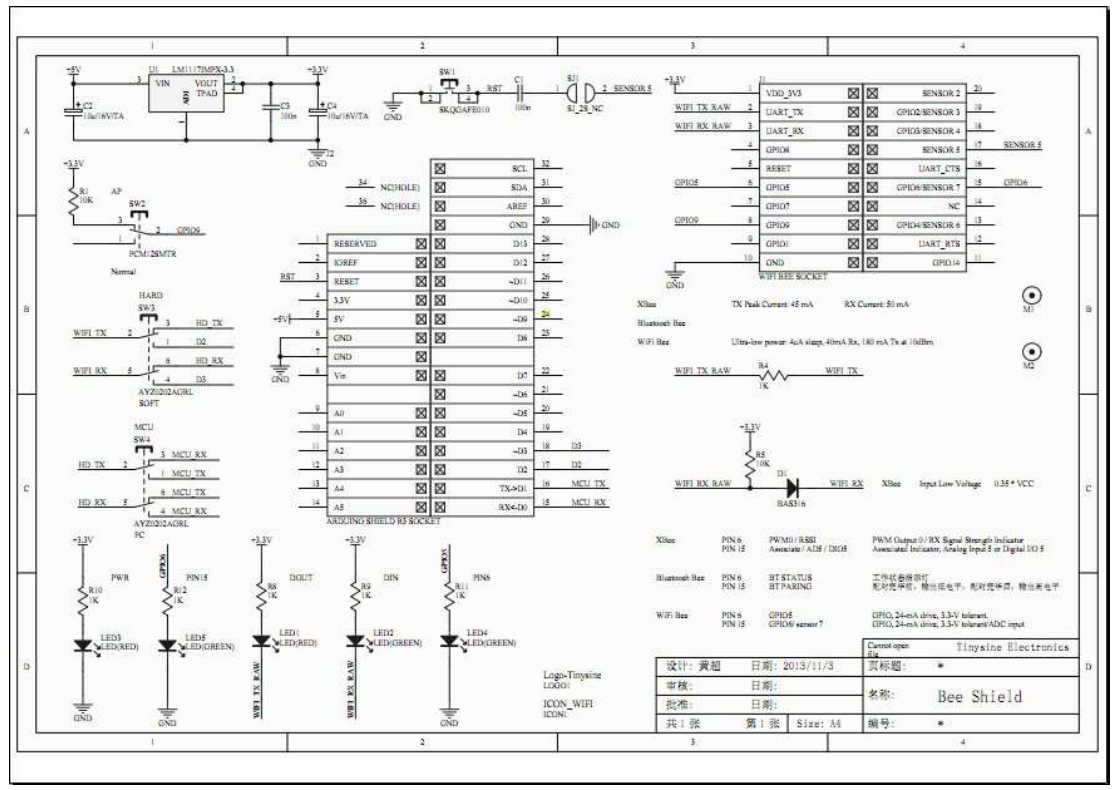
## Špecifikácia výrobcu

* Bluetooth chip: CSR BC417143
* Bluetooth protocol: Bluetooth Specification v2.0 + EDR
* USB Protocol: USB v1.1/2.0
* Operating frequency: 2.4 ~ 2.48GHz unlicensed ISM band
* Modulation: GFSK (Gaussian Frequency Shift Keying)
* Transmit Power: ≤ 4dBm, Class 2
* Transmission distance: 20 ~ 30m in free space
* Sensitivity: ≤-84dBm at 0.1% BER
* Transfer rate: Asynchronous: 2.1Mbps (Max) / 160 kbps; Synchronous: 1Mbps/1Mbps
* Safety features: Authentication and encryption
* Support profiles: Bluetooth serial port
* Serial port settings: 1200 ~ 1382400 / N / 8 / 1
* Baud rate default: 9600
* Pair code: 1234
* Input Voltage: +3.3 DC/50mA
* Operating temperature: -20 ℃ ~ +55 ℃
* Module Size: 32 × 24 × 9mm

## TinySine Bee Shield

TinySine Bee Shield je arduino shield [obr], ktorý umožňuje pripojiť XBee moduly ku ľubovolnej Arduino vývojovej doske. Shield poskytuje konverziu napäťových úrovní z 3.3 V na 5 V, obsahuje prepínače na presmerovanie sériovej komunikácie, regulátor na 3.3 V, niekoľko indikačných LED diód a tlačidlo Reset. Vhodnou kombináciu prepínačov MCU/PC a UART/Soft Serial vieme presmerovať dátovú komunikáciu pripojeného XBee modulu buď na piny arduina D0, D1, na ktorých je pripojený hardvarový UART mikropočítača, no zároveň aj USB to UART konvertor ktorým komunikuje MCU s pripojeným PC. Druhá možnosť je presmerovanie komunikácie na piny D2, D3 a v kombinácii so softwarovým UARTom môžeme bezproblémovo komunikovať s bluetooth modulom bez obmedzenia dátovej prevádzky na pinoch D0 a D1, ktoré budeme potrebovať na odlaďovanie programov.

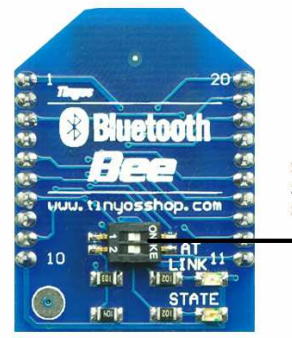
Prepínače by sa mali prestavovať pri vypnutom ziaradení (odpojený USB kábel).



## Aktivácia AT režimu

Modul HC-06 disponuje jedným sériovým komunikačným rozhraním ktorý slúži jednak na bluetooth komunikáciu (dátový režim) s MCU alebo na konfiguráciu modulu (AT režim). Tieto režimy sa prepínajú dvomi mikro prepínačmi na module Bluetooth Bee. Oba sú prepojené paralelne a pre aktiváciu AT režimu je potrebné prepnúť jeden z prepínačov do polohy ON (napravo). Pre aktiváciu dátového režimu musia byť oba prepínače v polohe OFF (naľavo) [obr].

Prepínače by sa mali prestavovať pri vypnutom ziaradení (odpojený USB kábel).



## Konfigurácia z vývojového prostredia Arduina

Pre konfiguráciu modulu prostredníctvom počítača je prepneme Bee Shield do takého režimu, kedy bude komunikačné sériové rozhranie bluetooth modulu (UART) prepojené priamo s hardvérovým UARTom Arduino dosky. Pri tomto zapojení využívame Arduino dosku iba ako konvertor USB na UART a všetky dáta vyslané z počítača pôjdu priamo na UART bluetooth modulu. Keďže UART na Arduino doske je zároveň pripojený na Mikrokontrolér Atmel, musíme zabezpečiť, aby program bežiaci na MCU neovplyvňoval dátovú komunikáciu. Keďže všetky komunikačné piny na doske arduino (nás zaujímajú iba D0 a D1) sú po naštartovaní MCU v režime vysokej impedancie, postačí nám napáliť prázdny program [obr], alebo môžeme odstrániť MCU z pätice.

C:\Data\Work\hc06\Pasted Graphic 2.tiff

Postup pre AT konfiguráciu bluetooth modulu cez vývojové prostredie Arduino:

* Odpojíme Bee Shield od Arduina
* Napálime prázdny program podľa obrázka [obr] alebo mechanicky odstránime MCU
* Odpojíme Arduino od USB
* Pripojíme Bee Shield
* Horný prepínač AT LINK na Bluetooth Bee module nastavíme do polohy ON (napravo) [obr]
* Prepínač MCU/PC prepneme do pozície PC
* Prepínač UART/Soft serial prepneme do pozície UART
* Prepojíme Arduino s PC prostredníctvom USB kábla
* Po zapnutí bliká indikačná dióda na Bluetooth Bee module ako aj na Bee Shielde s periódou asi 4 sekundy, toto znamená že modul je prepnutý do AT režimu
* V prostredí arduino zapneme Serial Monitor (Tools -> Serial monitor)
* Nastavíme prenosovú rýchlosť 38400 baudov a automatické odosielane CR/LF s každou správou
* Niekoľko krát odošleme správu “AT” vypísaním “AT” do editboxu v okne Serial monitor a potvrdením klávesy enter, po niekoľkých pokusoch sa vo výstupe objaví správa „OK“. Touto správou bluetooth modul potvrdzuje, že úspešne prijal srávu AT
* Modul je pripravený na konfiguráciu a prijímanie ďalších AT správ
* Vyskúšajme si dva AT príkazy:
  + „AT+UART?“ – týmto si vyžiadame konfiguráciu UARTU v dátovom režime, odpoveď by mala byť “+UART:9600,0,0”
  + “AT+NAME?” – odpoveďou na túto správu je názov bluetooth zariadenia, pod ktorým bude vystupovať v bezdrôtovej sieti

Po pripojení napájania modulu trvá niekoľko sekúnd kým začne reagovať a zároveň sa mohli dostať na jeho sériovú linku dáta ktoré odoslalo Arduino počas jeho bootovania. Následok tohoto chovania je ten, že na prvé odoslanie príkazu AT odpovedá bluetooth modul chybovou správou ERROR(0).

Pri poruche alebo odpojení USB kábla je potrebné zatvoriť okno Serial Monitor a po opätovnom pripojení ho znova otvoriť.

Pre vyšší komfort pri konfigurácii môžeme okrem interného Serial Monitora v Arduine použiť aj špecializovaný software na komunikáciu s terminálmi:

* Putty (Windows/Max/Linux)
* CoolTerm (Mac)
* Príkaz “screen /dev/tty…” na unixe/linuxe

C:\Data\Work\hc06\Pasted Graphic 4.tiff

## Vybrané AT Príkazy

|  |  |  |
| --- | --- | --- |
| **Príkaz** | **Odpoveď** | **Popis** |
| AT+UART? | +UART:9600,0,0 | Zistenie nastavenia UARTu v dátovom režime |
| AT+UART=9600,0,0 | OK | Nastavenie UARTu dátového režimu, čiarkou oddelený údaj o baudrate (prenosová rýchlosť, počet bitov za sekundu), údaj o počte stopbitov, a typ parity. Pozri poznámky za tabuľkou |
| AT+NAME? | +NAME:BluetoothBee | Vyžiadanie názvu modulu |
| AT+NAME=MojeMeno | OK | Nastavenie názvu |
| AT+PASSWD=1234 | OK | Nastavenie párovacieho pin kódu |
| AT+ROLE? | 0 | Zistenie úlohy modulu (0=slave, 1=master, 2=slave-loop) |
| AT+VERSION? | +VERSION:2.0-20100601 | Verzia firmwaru bluetooth modulu |
| AT+ORGL | OK | Reset všetkých nastavení na factory defaults |
| AT+ADDR? | +ADDR:13:12:304374 | MAC adresa zariadenia |
| AT+CLASS? | +CLASS:0 | 32 bitové číslo charakterizujúce kategóriu zariadenia (bt myš, audio headset…) |
| AT+INIT | OK | Zapnutie bluetooth rádia |
| AT+STATE? | +STATE:INITIALIZED | Vyžiadanie stavu zariadenia: initialized, ready, pairable, paired, inquiring, connected, disconnected, unknown |
| AT+INQ | OK + zoznam zariadení | Spusti scanovanie a vypisuj zoznam nájdených zariadení |
| AT+INQC | OK | Ukonči scanovanie |
| AT+PAIR=<devid>,<t> | OK | Spáruj sa so zariadením <dev> s časovým limitom <t>, časový limit je na vykonanie autentifikácie |
| AT+LINK=<devid> | OK | Spáruj sa so zariadením, s ktorým sme už boli autentifikovaný |
| AT+DISC | OK | Odpoj sa od zariadenia |

Ďalšie AT príkazy: AT+IAC, AT+INQM, AT+PSWD, AT+CMODE, AT+BIND, AT+MPIO, AT+IPSCAN, AT+SNIFF, AT+SENM, AT+RMSAD, AT+FSAD, AT+ADCN, AT+MRAD, AT+INIT, AT+ENSNIFF, AT+EXSNIFF, AT+POLAR, AT+PIO

Poznámky ku príkazom:

**AT+UART**

Podporované prenosové rýchlosti: 4800\*\*, 9600\*\*, 19200\*, 38400\*, 57600, 115200, 230400, 460800, 921600, 1382400

Počet stopbitov: 0=1 stop bit, 2=2 stop bity

Typ parity: 0=žiadna, 1=nepárna (odd), 2=párna (even)

\*\* = odporúčaná voľba pri komunikácii s knižnicou SoftSerial

\*\*, \* = odporúčaná voľba pri kominukácii s hardwarovým UARTom na doske Arduino Uno

**AT+ROLE?**

Modul môže pracovať v jednom z troch režimov. Najbežnejší z nich je režim slave (ROLE 0), kedy modul čaká, kým sa na neho niekto bezdrôtovo pripojí – napr. telefón alebo počítač. Režim master (ROLE 1) inicializuje spojenie automaticky, ak sa v jeho blízkosti objaví zariadenie s ktorým sa má spárovať. Pre automatické spárovanie treba nastaviť adresu tohto cieľového zariadenia (AT+RNAME) a jeho párovací pin kód

**AT+ORGL**

Týmto príkazom resetujeme všetky nami nastavené zmeny v konfigurácii modulu, príkaz nastaví modul nasledovne:

* Device class: 0
* Inquiry code: 0x009e8b33
* Device mode: slave
* Binding mode: SPP
* Serial port: 38400, 1 stop bit, no parity
* Pairing code: 1234
* Device name: H-C-2010-06-01

**AT+CLASS**

Trieda BT zariadenia (CLASS) charakterizuje zariadenie z hľadiska služby ktorú poskytuje (Major service class) [link]:

* Sieťové zariadenie
* Renderovacie zariadenie: Tlačiareň/Reproduktor
* Capturing zariadenie: Scanner/Mikrofón
* Transfer objektov: Virtuálne vizitky
* Zvukové zariadenie: Reproduktor, Mikrofón, Handsfree
* ...

A z hľadiska druhu zariadenia (Major device class):

* Počítač
* Telefón
* LAN/Access point
* Audio/Video
* Periféria: Myš, klávesnica, joystick
* ...

Niekedy mobilný telefón odmietne spárovať sa so zariadením, ktoré má nastavené device CLASS=0, vtedy si treba vygenerovať nejaký zmysluplný kód s pomocou stránky [link]

**AT+INQ**

Pre scanovanie BT zariadení v okolí môžeme zavolať nasledujúce príkazy:

* AT+INIT
* AT+IAC=9E8B33
* AT+CLASS=0
* AT+INQM=1,9,48
* QT+INQ

A pre ukončenie scanovania:

* AT+INQC

## Chybové kódy

Modul reaguje na AT príkazy najčastejšie odpoveďou “OK”, chybové hlásenia sú v tvare “ERROR:(17)”, a číselné označenie chyby môžeme dekódovať podľa nasledujúcej tabuľky:

C:\Data\Work\hc06\Pasted Graphic.tiff

# Dátový režim

Ak máme modul úspešne nakonfigurovaný, prepneme zariadenie do dátového režimu. V tomto režime je BT modul pripravený na spárovanie a obojsmernú komunikáciu s počítačom, resp. telefónom, alebo druhým BT modulom. Z praktických dôvodov budeme na strane Arduina používať softvérovú implementáciu UARTU (SoftwareSerial), ktorá bude bežať na pinoch D2 a D3. Piny D0 a D1 si ponecháme na výpis ladiacich správ prostredníctvom funkcie Serial.Print.

Pre nastavenie dátového režimu postupujeme nasledovne:

* Prepínač AT LINK v polohe OFF (naľavo)
* Prepínač MCU/PC v polohe MCU???
* Prepínač UART/Soft serial v polohe Soft serial
* Pripojíme USB kábel, Indikačná dióda P6 bliká približne dva razy za sekundu

Aby sme overili funkčnosť modulu, necháme preposielať všetky prijaté dáta z Bluetooth modulu na hardwarový UART port. Všetky prijaté dáta z BT modulu budú následne prenesené cez USB kábel do počítača a môžeme ich vidiet v integrovanom Serial Monitore v Arduino IDE.

Pre preposielanie dát, bez znalosti nastaveného baudratu môžeme použiť nasledujúci program:

void setup()

{

pinMode(0, INPUT); // Hw serial RX

pinMode(1, OUTPUT); // Hw serial TX

pinMode(2, INPUT); // Sw serial RX

pinMode(3, OUTPUT); // Sw serial TX

}

void loop()

{

while (1)

{

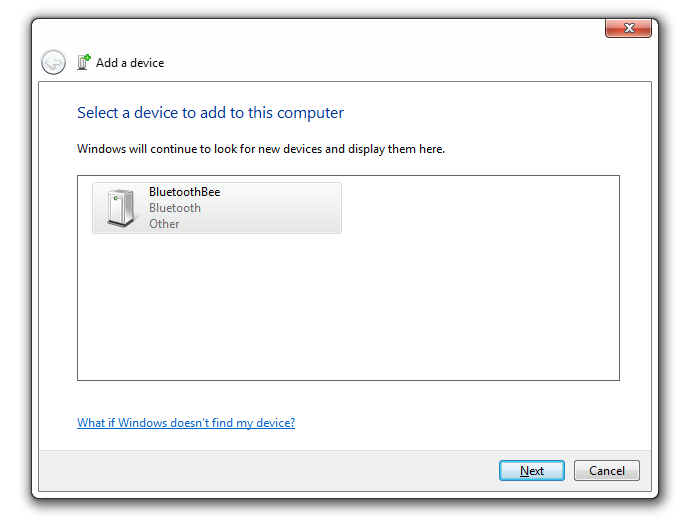
digitalWrite(1, digitalRead(2)); // Hw.Tx <- Sw.Rx

digitalWrite(3, digitalRead(0)); // Sw.Tx <- Hw.Rx

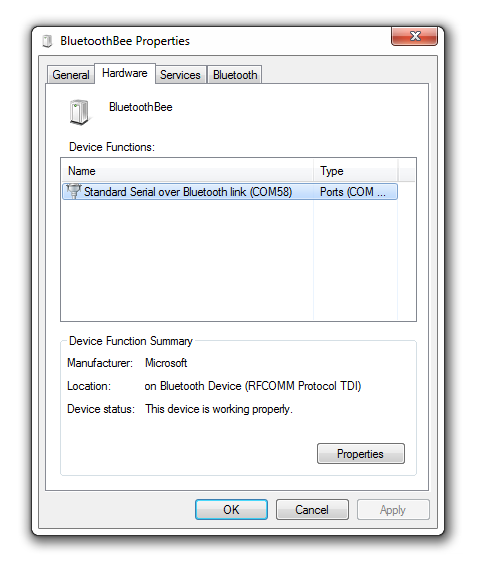
}

}

Na strane počítača vyhľadáme okolité bluetooth zariadenia (Na windowse: Devices and printers -> Add device) modul by sme mali vidieť pod názvom ktorý sme zadali počas konfigurácie v predošlej časti



Po spárovaní, zadaní pin kódu a pridelení virtuálneho sériového portu, sa môžeme na zariadenie pripojiť z utility putty, coolterm, alebo môžeme použiť príkaz screen. Ak potrebujeme zistiť sériový port, pod ktorým naše zariadenie vystupuje, pravým klikom na ikonu zariadenia v okne Devices and printers získame údaje o našom zariadení:



Teraz môžeme použiť utilitu putty, screen, alebo coolterm na otestovanie komunikácie. Všetky odoslané znaky z terminálu na počítači by sa mali objaviť v okne Serial Monitor prostredia Arduino. Nesmieme zabudnúť na nastavenie správneho baudratu v okne Serial Monitor (9600).

Počas testovania môže nastať situácia, že komunikácia funguje zdanlivo iba v jednom smere. Pri vysielaní údajov zo Serial Monitoru sa tieto neobjavia okamžite v okne terminálu, tento jav súvisí pravdepodobne s cachovaním sériovej linky, alebo s riadením toku (flow control). Treba znovu prepnúť modul do AT režimu a znova nastaviť baudrate (AT+UART=9600,0,0).

Pre preposielanie dát, s použitím softwarovej emulácie UARTu môžeme použiť tento program:

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX

void setup()

{

Serial.begin(9600);

while (!Serial);

Serial.println("Ready!");

mySerial.begin(9600);

}

void loop()

{

if (mySerial.available())

Serial.write(mySerial.read());

if (Serial.available())

mySerial.write(Serial.read());

}

Program, ktorý každú sekundu odošle správu (používa časovač TimerOne):

#include <SoftwareSerial.h>

#include <TimerOne.h>

SoftwareSerial btSerial(2, 3); // RX, TX

volatile int nSecondsPassed = 0;

int nLastSecond = 0;

void setup()

{

Serial.begin(9600);

while (!Serial);

btSerial.begin(9600);

int nFreq = 1;

Timer1.initialize(1000000UL / nFreq);

Timer1.attachInterrupt(onTimerIsr);

}

void onTimerIsr()

{

nSecondsPassed++;

}

void onTimer()

{

Serial.print("Send...\r\n");

btSerial.print("Ahoj, preslo ");

btSerial.print(nSecondsPassed);

btSerial.print(" sekund!\r\n");

}

void loop()

{

if ( nLastSecond != nSecondsPassed )

{

nLastSecond = nSecondsPassed;

onTimer();

}

while (btSerial.available())

Serial.write(btSerial.read());

if (Serial.available())

btSerial.write(Serial.read());

}

Program na odosielanie teploty a vlhkosti (senzor DHT22) cez bluetooth:

#include <SoftwareSerial.h>

#include <TimerOne.h>

#include <DHT.h>

SoftwareSerial btSerial(2, 3); // RX, TX

volatile int nSecondsPassed = 0;

int nLastSecond = 0;

DHT sensorDht(10, DHT22);

void dhtInit ()

{

// DHT senzor na pinoch 11..8

pinMode(8, OUTPUT);

digitalWrite(8, LOW);

pinMode(9, INPUT);

pinMode(10, INPUT);

pinMode(11, OUTPUT);

digitalWrite(11, HIGH);

}

void setup()

{

Serial.begin(9600);

while (!Serial);

btSerial.begin(9600);

int nFreq = 1;

Timer1.initialize(1000000UL / nFreq);

Timer1.attachInterrupt(onTimerIsr);

dhtInit();

}

void onTimerIsr()

{

nSecondsPassed++;

}

void onTimer()

{

Serial.print("Send...\r\n");

btSerial.print("Ahoj, preslo ");

btSerial.print(nSecondsPassed);

btSerial.print(" sekund!");

btSerial.print(" dht:");

float h = sensorDht.readHumidity();

float t = sensorDht.readTemperature();

if ( isnan(h) || isnan(t) )

{

btSerial.print("error");

} else

{

btSerial.print("temp=");

btSerial.print(t);

btSerial.print(" humidity=");

btSerial.print(h);

}

btSerial.print("\r\n");

}

void loop()

{

if ( nLastSecond != nSecondsPassed )

{

nLastSecond = nSecondsPassed;

onTimer();

}

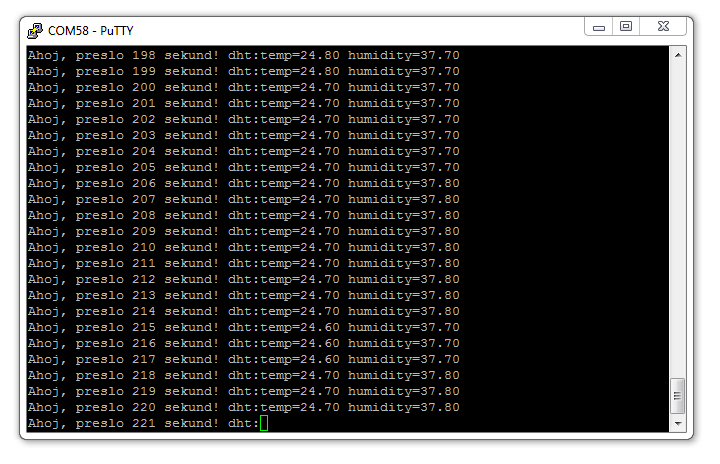
while (btSerial.available())

Serial.write(btSerial.read());

if (Serial.available())

btSerial.write(Serial.read());

}



Linky:

<https://en.wikipedia.org/wiki/XBee>

http://www.tinyosshop.com/index.php?route=product/product&product\_id=338

http://www.tinyosshop.com/datasheet/Bluetooth%20Bee%20manual.pdf

http://www.tinyosshop.com/index.php?route=product/product&product\_id=681

http://bluetooth-pentest.narod.ru/software/bluetooth\_class\_of\_device-service\_generator.html