Opis interfejsu programistycznego dla tinyBrd

wersja 1.0, data: sierpień 2015
TinyBrd: https://nettigo.pl/products/bezprzewodowy-sensor-tinybrd

API - Rabarbar (Python)

pyNRF

address - 3 elementowy typ bytes

Konstruktor:

Radio(address, channel=1, bus=0, csdevice=0, ce_pin=22)

Przygotowuje rasdio do działania
Argumenty:
address - adres pod którym modem będzie odbierał dane
channel - kanał po którym modemy mają się komunikować 1 - 125 (od 2,4 do 2,525 GHz)
bus - numer magistrali SPI
csdevice - numer sygnału wyboru urządzenia CS
ce_pin - numer pinu aktywacji nadawania/odbioru
Wartość zwracana:
obiekt typu Radio

Radio.write(address, data)

Wysyła dane do odbiornika
Argumenty:
address - adres odbiornika
data - dane do wysłania - typu bytes

Radio.read()

Odczytuje odbrane dane
Wartość zwracana:
odebrane dane - typu bytes

Radio.available()

Zwraca czy odebrano jakieś dane
Wartość zwracana:
True - odebrano
False - nie odebrano

Radio.flush(blocking=BLOCK)

Zwraca czy dane dotarły do odbiorcy
Argumenty:
blocking - ustala czy funkcja ma blokować program do chwili potwierdzenia odebrania danych
tinybrd.BLOCK - blokuje
tinybrd.NONBLOCK - nie blokuje

Wartość zwracana
tinybrd.WAIT - dane nie zostały jeszcze wysłane
tinybrd.SENT - dane dotarły do odbiorcy
tinybrd.LOST - dane nie dotarły do odbiorcy

Radio.off()

Wyłacza modem

Dodatkowe zaawansowane funkcje

Radio.device.speed_power(speed, power);

Ustawia prędkość przesyłania danych i moc nadajnika
Argumenty:
speed
- 0 - 1 Mbps,
- 1 - 2 Mbps,
- 2 - 250 kbps

power
- 0 - -18 dBm/7 mA
- 1 - -12 dBm/7,5 mA
- 2 - -6 dBm/9 mA
- 3 - 0 dBm/11,3 mA

Radio.device.tx_data_lost();

Zwraca krotkę z danymi o zgubionych pakietach danych
Wartość zwracana:
pierwsza wartość - ilość pakietów danych, które nie dotarły do odbiornika
druga wartość - ilość powtórek wysyłania zanim pakiety dotarły do odbiornika

Radio.device.rx_signal_strength();

Zwraca siłę odebranego sygnału z pakietem danych
Wartość zwracana:

1 - jeśli odebrany sygnał ma większą energię niż -64 dBm
0 - jeśli odebrany sygnał jest słabszy niż -64 dBm

API TinyBrd (C++)

Radio.h

address - tablica typu byte o rozmiarze 3 elementów

Radio.begin(address, channel);

Przygotowuje radio do działania.
Argumenty:
address - adres na jaki radio będzie odbierało dane
channel - kanał radiowy do komunikacji między modemami od 0 do 125 (od 2,4 do 2,525 GHz)

Radio.write(address, struct);

Wysyła strukturę do odbiornika
Argumenty:
address - adres odbiornika
struct - obiekt struktury do przesłania

Radio.write(address, *data, size);

Wysyła dane na które wskazuje wskaźnik do odbiornika.
Argumenty:
address - adres odbiornika
data - wskaźnik na dane
size - ilośc bajtów do przesłania

Radio.flush(blocking=RADIO_BLOCK);

Sprawdza czy dane dotarły do odbiorcy
Argumenty:
blocking - ustala czy funkcja ma blokować program do czasu potwierdzenia odebrania danych
RADIO_BLOCK - funkcja blokuje program
RADIO_NONBLOCK - funkcja nie blokuje programu

Wartość zwracana:
RADIO_WAITS - dane jeszcze nie zostały przesłane
RADIO_SENT - dane zostały odebrane przez odbiornik
RADIO_LOST - dane nie dotarły do odbiorcy

Radio.available();

Zwraca ilość danych odebranych przez radio

Wartość zwracana:

0 - radio niczego nie odebrało

n>0 - radio odebrało dane

Radio.read(*data);

Odczytuje odebrane dane

Argumenty:

data - wskaźnik do miejsca gdzie mają być zapisane odebrane dane

Radio.off();

Wyłącza radio i oszczędza prąd

Dodatkowe zaawansowane funkcje

Radio.device.speedPower(speed, power);

Ustawia prędkość przesyłania danych i moc nadajnika

Argumenty:

speed

0 - 1 Mbps,

1 - 2 Mbps,

2 - 250 kbps

power

0 - -18 dBm/7 mA

1 - -12 dBm/7,5 mA

2 - -6 dBm/9 mA

3 - 0 dBm/11,3 mA

Radio.device.txDataLost();

Zwraca strukturę z danymi o zgubionych pakietach danych

Wartość zwracana:

struktura RadioLost z polami:

lost - ilość pakietów danych, które nie dotarły do odbiornika

retr - ilość powtórek wysyłania zanim pakiety dotarły do odbiornika

Radio.device.rxSignalStrength();

Zwraca siłę odebranego sygnału z pakietem danych

Wartość zwracana:

1 - jeśli odebrany sygnał ma większą energię niż -64 dBm

0 - jeśli odebrany sygnał jest słabszy niż -64 dBm

Battery.h

batteryRead();

Odczytuje napięcie zasilania płytki

Wartość zwracana:

Napięcie baterii w mV

sleep(sleepTime);

Przełącza procesor w tryb oszczędzania energii, wyłącza wszystkie urządzenia i wyłącza procesor na określony czas.

Argumenty:

sleepTime - czas wyłączenia procesora w milisekundach z dokładnością do 15 ms.

Storage.h

storageWrite(address, struct);

Zapisuje dane do pamięci nieulotnej/trwałej (EEPROM)

Argumenty:

adress - adres w pamięci w bajtach

struct - obiekt struktury do zapisania w pamięci

Wartość zwracana:

rozmiar zapisanej strukltury w bajtach

storageRead(address, struct);

Odczytuje dane z pamięci trwałej (EEPROM)

address - adres w pamięci w bajtach

struct - obiekt struktury do odczytania z pamięci

flashRead(address, struct);

Odczytuje dane z pamięci Flash

address - wskaźnik do danych w pamięci Flash

struct - obiekt struktury do odczytania z pamięci