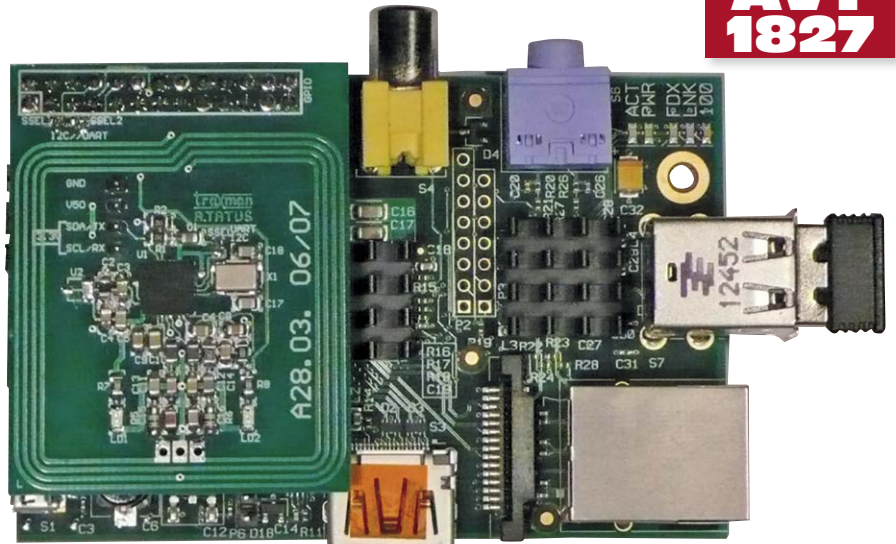


# RaspbPI\_NFC – płytki czytnika RFID dla Raspberry Pi i nie tylko

Przedstawiona płytki umożliwia rozszerzenie funkcjonalności Raspberry Pi o możliwość odczytu kart RFID w standardzie Mifare.

Płytki wymaga Raspberry Pi Rev2. o rozmieszczeniu sygnałów GPIO-P1 przedstawionym w tabeli 1. Dla potrzeb współpracy domyślnie jest używany interfejs UART, wspierany przez bibliotekę *libnfc*. Możliwe jest też zastosowanie interfejsu I<sup>2</sup>C. Dla ułatwienia współpracy z innymi komputerkami (oprócz Raspberry Pi) płytki ma złącze EH w standardzie Arduino,

Moduł jest oparty o popularny układ PN532 firmy NXP. Odpowiada on za kompleksową obsługę kart zbliżeniowych Mifare, umożliwiając szybkie i łatwe wykonanie urządzeń NFC. Schemat blokowy układu pokazano na rysunku 1.



Schemat modułu NFC przedstawia rysunek 2. Moduł nie odbiega od noty aplikacyjnej NXP. Rezonator kwarcowy X1 jest

elementem układu zegarowego, rezystory R3 i R6, kondensatory C6, C7, C11 i C16 dopasowują antenę i stanowią filtr EMC. Pozostałe

# MINIPROJEKTY

Tabela 1. Przyporządkowanie sygnałów GPIO1 – P1 (na szaro zaznaczono piny używane przez RasbpPI\_NFC)

P1 GPIO			
Pin	Funkcja		Pin
P1-01	3.3V	5V	P1-02
P1-03	GPIO2(SDA)	5V	P1-04
P1-05	GPIO3(SCL)	GND	P1-06
P1-07	GPIO4	GPIO14(TXD)	P1-08
P1-09	GND	GPIO15(RXD)	P1-10
P1-11	GPIO17	GPIO18	P1-12
P1-13	GPIO27	GND	P1-14
P1-15	GPIO22	GPIO23	P1-16
P1-17	3.3V	GPIO24	P1-18
P1-19	GPIO10(MOSI)	GND	P1-20
P1-21	GPIO9(MISO)	GPIO25	P1-22
P1-23	GPIO11(SCLK)	GPIO8(CE0)	P1-24
P1-25	GND	GPIO7(CE1)	P1-26

W ofercie AVT\*  
AVT-1827 A

Wykaz elementów:

**Rezystory:** (SMD 0805)

- R1, R2: 3,3 kΩ
- R3, R7, R8: 1 kΩ
- R4: 1,6 kΩ
- R5, R6: 1,5 Ω
- R9: 10 kΩ

**Kondensatory:** (SMD 0805, ceram.)

- C1, C3, C5, C6, C8, C10: 0,1 μF
- C2, C4, C9: 10 μF
- C7: 1 nF (NP0)
- C11, C12: 220 pF (NP0)
- C13, C14: 22 pF (NP0)
- C15, C16: 100 pF (NP0)
- C17, C18: 18 pF (NP0)

**Półprzewodniki:**

- LD1, LD2: dioda LED SMD
- U1: PN532 (HVQFN40)

**Inne:**

- GPIO: złącze IDC26 żeńskie
- I2C: złącze EH kątowe (opcja)
- L1, L2: 560 nH (SMD 0805)
- SSEL, SSEL1, SSEL2: zwora
- X1: 27,12 MHz (kwarc SMD 5 mm×3,2 mm)

Dodatkowe materiały na FTP:

<ftp://ep.com.pl>, user: 76305, pass: 8741rnfv

• wzory płytek PCB

Projekty pokrewne na FTP:

(wymienione artykuły są w całości dostępne na FTP)

AVT-5459 RaspbPI GSM Płytką z modemem GSM dla Raspberry Pi (EP 7/2014)

AVT-5431,-32,-33 Moduły rozszerzeń dla Raspberry Pi (4) – RaspbPI\_LCD, RaspbPI\_Relay, RaspbPI\_LED8\_PWM\_Expander (EP 1/2014)

AVT-5412,-13,-14 Moduły rozszerzeń dla Raspberry Pi (3) – RaspbPI\_DIO16, RaspbPI\_HUB, RaspbPI\_DCM (EP 9/2013)

\* Uwaga:

Zestawy AVT mogą występować w następujących wersjach: AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.

AVT xxxx A płytką drukowaną PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.

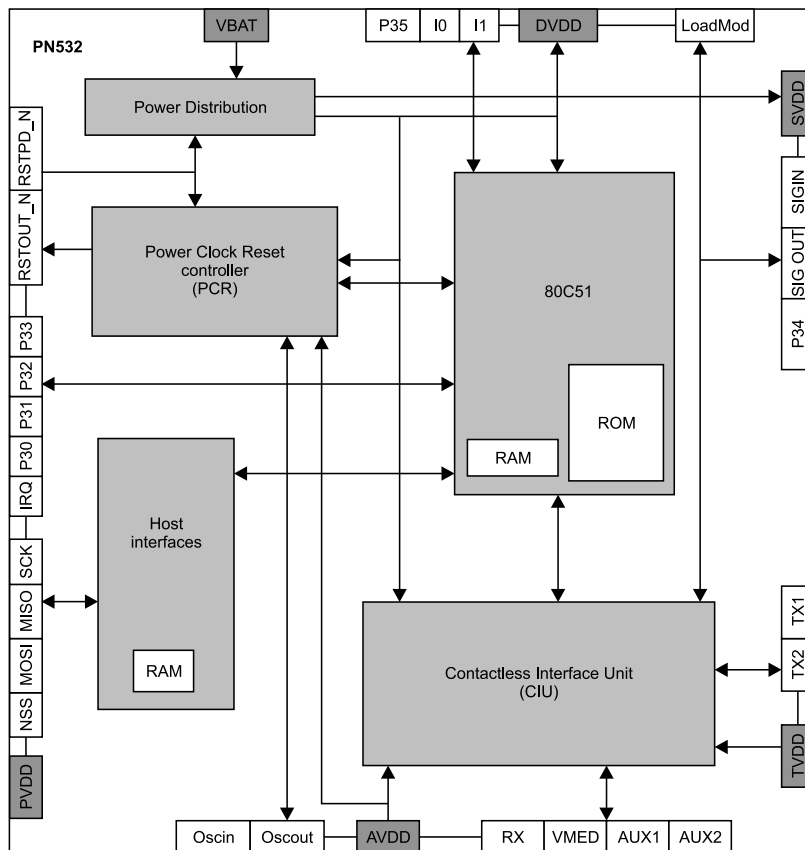
AVT xxxx A+ płytką drukowaną i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.

AVT xxxx B płytką drukowaną (lub płytki) oraz komplet elementów wymieniony w załączniku pdf

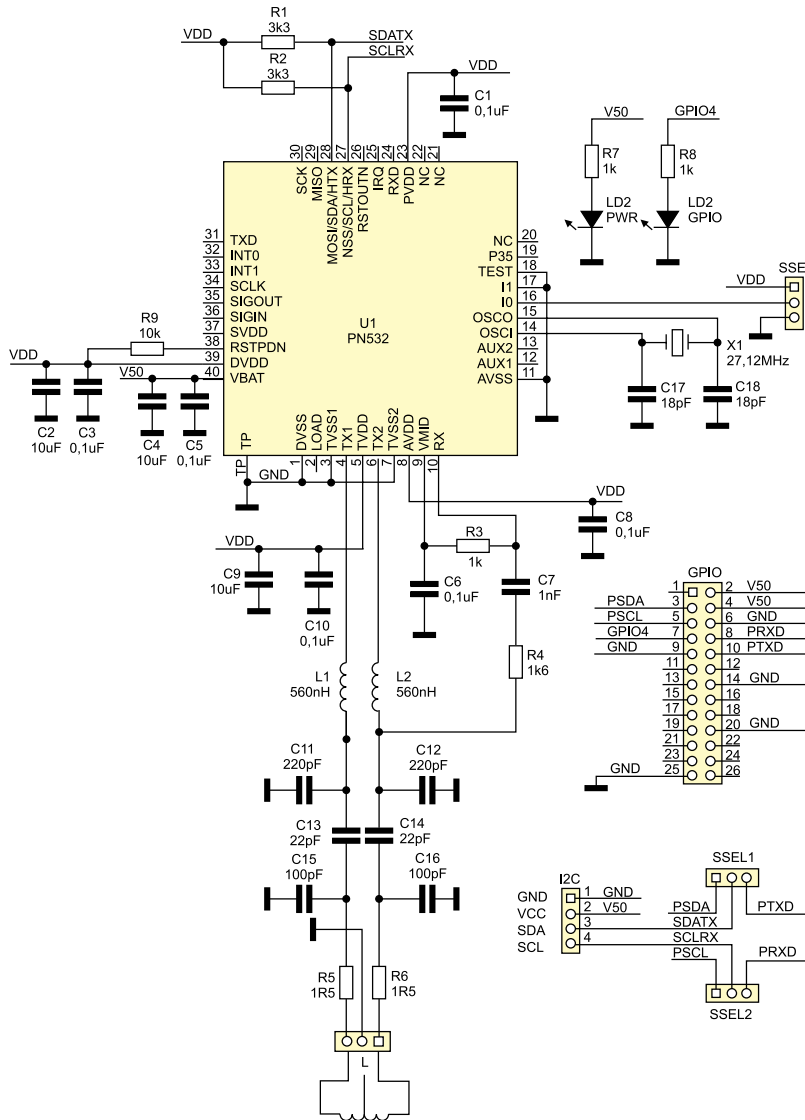
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wstawiane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf

AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)

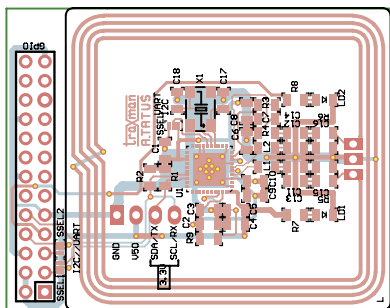
Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>



Rysunek 1. Schemat blokowy układu PN532 (za nota NXP)



Rysunek 2. Schemat ideowy modułu RasbpI\_NFC



Rysunek 3. Rozmieszczenie elementów modułu NFC

kondensatory filtrują zasilanie. Antenę L wykonano na płytce drukowanej. Ułatwia to uruchomienie modułu uwalniając od konieczności nawijania cewek antenowych. Niestety, jest okupione nieco mniejszą dobrocią ograniczającą zasięg do ok. 3 cm w przypadku kart i 1 cm w przypadku żetonów.

Do komunikacji z Raspberry jest wykorzystywany interfejs UART. Aby nie ograniczać użyteczności pozostawiono także możliwość komunikacji poprzez PC. Wyboru interfejsu dokonuje się za pomocą zwór SSEL/SSEL1/SSEL2, zgodnie z opisem umieszczonym na płytce. Dodatkowo, interfejs komunikacyjny wyprowadzono na złącze „I2C” zgodne z Arduino.

Moduł jest zasilany z 5 V. Należy pamiętać o konieczności dopasowania poziomów logicznych interfejsu w zależności od współpracującego układu, gdyż PN532 pracuje z napięciem 3,3 V. Moduł ma sygnalizację załączenia zasilania za pomocą diody świecącej LD1 oraz diodę LD2 sterowaną z GPIO4, która może być użyta do sygnalizacji działania NFC.

Układ zmontowano na niewielkiej, dwustronnej płytce drukowanej. Jej schemat montażowy pokazano na **rysunku 3**. Montaż nie

wymaga opisu, należy tylko zwrócić uwagę na prawidłowe przylutowanie pada termicznego U1 oraz odpowiednie wlutowanie zwór SSEL/SSEL1/SSEL2 w zależności od używanego interfejsu komunikacyjnego (domyślnie UART).

Aby w praktyce najszybciej sprawdzić działanie modułu konieczne są drobne zmiany konfiguracji PI oraz instalacja biblioteki libnfc. W pierwszej kolejności musimy uzyskać dostęp do portu szeregowego, który jest domyślnie zablokowany przez terminal SSH. W tym celu należy otworzyć do edycji plik `cmdline.txt` (`$ sudo nano /boot/cmdline.txt`) i usunąć wpisy dotyczące konsoli `console=ttyAMA0,115200 kgdboc=ttyAMA0,115200`. Następnie należy otworzyć do edycji plik `initab` (`$ sudo nano /etc/initab`) i opatrzyć komentarzem linię `TO:23respawn:/sbin/getty -L ttyAMA0 115200 vt100` (`#TO:23respawn:/sbin/getty -L ttyAMA0 115200 vt100`). Po wprowadzeniu zmian, należy zrestartować Raspberry.

Po tej operacji port szeregowy UART jest dostępny dla innych aplikacji. Aby wykorzystać płytkę NFC jest konieczna jeszcze instalacja darmowej biblioteki `libnfc` (najnowsza wersja 1.7.1), do pobrania z: <http://goo.gl/3qlidU>. Bibliotekę należy pobrać i rozpakować do folderu domowego Pi `/home/pi/libnfc-1.7.1`. Następnie należy stworzyć katalog dla plików konfiguracyjnych `sudo mkdir /etc/nfc` && `sudo mkdir /etc/nfc/devices.d/` oraz edytować zawartość `sudo nano /etc/nfc/devices.d/pn532_via_uart.conf` uzupełniając zawartość pliku:

```
## Typical configuration file for
PN532 board
name = „NFC PN532 UART”
connstring = pn532_uart:/dev/
ttyAMA0
allow_intrusive_scan = true
```

```
log_level = 3
Następnie należy wydać polecenia:
cd /etc/nfc
sudo nano libnfc.conf
i uzupełnić zawartość pliku:
# Allow device auto-detection
(default: true)
# Note: if this auto-detection is
disabled, user has to set manually
a device
# configuration using file or
environment variable
allow_autoscan = true
# Allow intrusive auto-detection
(default: false)
# Warning: intrusive auto-
detection can seriously disturb
other devices
# This option is not recommended,
user should prefer to add manually
his device.
#allow_intrusive_scan = false
# Set log level (default: error)
# Valid log levels are (in order
of verbosity): 0 (none), 1
(error), 2 (info), 3 (debug)
# Note: if you compiled with
--enable-debug option, the default
log level is „debug”
#log_level = 1
# Manually set default device (no
default)
# To set a default device, you
must set both name and connstring
for your device
# Note: if autoscan is enabled,
default device will be the first
device available in device list.
#device.name = „microBuilder.eu”
#device.connstring = „pn532_uart:/
dev/ttyAMA0”
```

Po zapisaniu plików konfiguracyjnych jest konieczne skompilowanie biblioteki z wcześniej ustalonymi parametrami. Należy wrócić do katalogu `/home/pi/libnfc-1.7.1` i wydać polecenie konfigurujące driver dla UART `sudo ./configure --sysconfdir=/etc --prefix=/usr --with-drivers=pn532_uart` oraz uruchomić `sudo make clean all` i `sudo make install all`. Po pomyślnym zakończeniu pracy kompilatora wracamy do podkatalogu biblioteki `/home/pi/libnfc-1.7.1` (`cd /utils`) i wydajemy polecenie `sudo ./nfc-list`. Jeżeli wszystkie wcześniejsze operacje przebiegły pomyślnie i moduł jest zmontowany poprawnie, czytnik kart zostanie zainicjowany, następnie przechodzimy do katalogu `cd ../examples` i wydajemy polecenie `sudo ./nfc-poll`. Program będzie oczekiwał na kartę, po zbliżeniu zostanie odczytany jej numer. Przykładowy odczyt pokazano na **rysunku 4**. Jeżeli wszystko funkcjonuje poprawnie, czytnik może być użyty we własnej aplikacji.

```
pi@raspberrypi: ~/libnfc-1.7.1/examples
File Edit Tabs Help
pi@raspberrypi ~ $ cd ./libnfc-1.7.1/utils
pi@raspberrypi ~/libnfc-1.7.1/utils $ sudo ./nfc-list
/home/pi/libnfc-1.7.1/utils/.libs/lt-nfc-list uses libnfc 1.7.1
NFC device: pn532_uart:/dev/ttyAMA0 opened
pi@raspberrypi ~/libnfc-1.7.1/examples $ cd ../examples
pi@raspberrypi ~/libnfc-1.7.1/examples $ sudo ./nfc-poll
/home/pi/libnfc-1.7.1/examples/.libs/lt-nfc-poll uses libnfc 1.7.1
NFC reader: pn532_uart:/dev/ttyAMA0 opened
NFC device will poll during 30000 ms (20 pollings of 300 ms for 5 modulations)
ISO/IEC 14443A (106 kbps) target:
  ATQA (SENS_RES): 00 04
  UID (NFCID1): 3d d7 93 19
  SAK (SEL_RES): 08
nfc_initiator_target_is_present: Target Released
Waiting for card removing...done.
pi@raspberrypi ~/libnfc-1.7.1/examples $ sudo ./nfc-poll
/home/pi/libnfc-1.7.1/examples/.libs/lt-nfc-poll uses libnfc 1.7.1
NFC reader: pn532_uart:/dev/ttyAMA0 opened
NFC device will poll during 30000 ms (20 pollings of 300 ms for 5 modulations)
ISO/IEC 14443A (106 kbps) target:
  ATQA (SENS_RES): 00 04
  UID (NFCID1): 8d cb 83 cd
  SAK (SEL_RES): 08
nfc_initiator_target_is_present: Target Released
Waiting for card removing...done.
pi@raspberrypi ~/libnfc-1.7.1/examples $
```

Rysunek 4. Test modułu RasbPI\_NFC

Adam Tatuś, EP