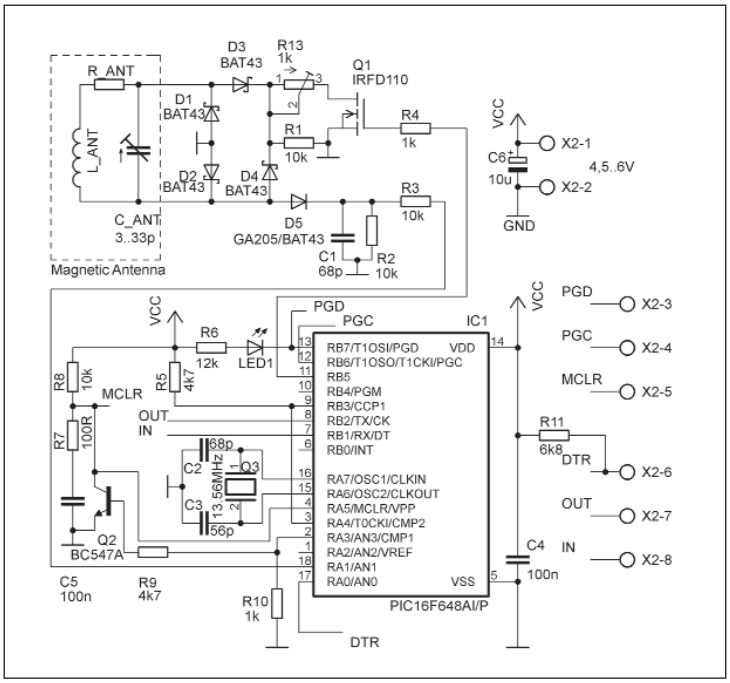
PicNic pro RFID-KV

Při experimentech s čipy RFID v pásmu KV začneme brzy intenzivně postrádat přípravek schopný pracovat jednak jako emulátor transpondéru, jednak jako pasivní odposlech alespoň dat vysílaných „čtečkou“ do nějaké karty. Konkrétně taková potřeba vzniká například při penetračních testech, kdy chceme demonstrovat nevhodnost spoléhání se jen na číslo karty Mifare (viz ST 2/2007 v [5]). Málokterý manažer nám uvěří, dokud si třeba v závodní kantýně nekoupíme slušný oběd na jeho účet. Je to s podivem, ale sehnat něco takového coby hotový nástroj je pro běžného studenta, pedagoga či bezpečnostního referenta prakticky nepřekonatelný problém. Komerční zařízení jsou poněkud drahá (jednotky až desítky tisíců Euro) a na volně dostupných zapojeních se zase neblaze podepisuje jistá megalomanie jejich konstruktérů. Výrobou mnohavrstvých plošných spojů počínaje, přes programování hradlových polí a programováním komplikovaných procesorů a operačních systémů konče, nás jejich příprava spolehlivě odláká daleko od původního záměru analyzovat nějaký čip RFID. Proto jsme navrhli vlastní zařízení, kterému pro změnu dominuje snaha o extrémně jednoduchou konstrukci.

Popis zapojení

Již v názvu přípravku je vyjádřeno základní návrhové paradigma – použít kromě vybraného řadiče PIC co nejméně okolních prvků (tedy PicNic = PIC a „Nic“). Využít tento široce známý a oblíbený procesor se rozhodl i autor diplomové práce [4]. Na náš vkus ale stále potřebuje příliš mnoho okolních prvků, takže ten správný „piknik“ to ještě není (nehledě na autorovu volbu zbytečně komplikovaného vývojového prostředí).

V pilotní verzi určené zejména k experimentům s útoky na čipy Mifare (viz příští díly) jsme se rozhodli pro podporu standardu ISO 14443A. Věříme, že s drobnými úpravami v zapojení a zdrojových kódech (hlavně tam) je piknikový přístup aplikovatelný i na ISO 14443B a ISO 15693. Zapojení umožňuje připojení klasického sériového programátoru bez nutnosti demontáže řadiče, takže ladění obslužného kódu je poměrně snadné. Konkrétně jsme zvolili procesor PIC16F648A [3], v případě větších nároků na paměť by nemělo být těžké najít jiného vhodného kandidáta na-



Obr. 1 Schéma PicNicu verze 1.0

bízejícího alespoň tytéž vestavěné periferie. Právě díky bohaté výbavě čipu řadiče jsme mohli piknikové paradigma úspěšně realizovat. Nakreslením schématu zahrnujícího všechny skutečně použité komponenty dostaneme poněkud složitější zapojení, než je na *obr. 1*, které se potom nepřekvapivě blíží práci [4]. S ohledem na prostor ho zde uvádět nebudeme, nicméně podle zdrojového kódu není těžké si ho dovodit.

Magnetická anténa PicNicu je tvořena vlastní cívkou, sériovým rezistorem a ladicím kondenzátorem. Vzhledem k tomu, že u použitého typu antény velmi záleží na tom, jak se zrovna „povede“, uvedeme zde raději obecný konstrukční návod. Několik antén jsme podle něj postavili a fungovaly prakticky na první zapojení. Z hlediska elektromagnetického pole je vhodné cívku antény konstruovat jako několik závitů v rovině kolem plochy zhruba odpovídající běžné čipové kartě. Z hlediska obvodového nás zajímá hlavně její indukčnost, přehled užitečných vzorců viz [1]. Tu musíme udržet v řádu nejvýše desítek μH, abychom byli schopni nastavit příslušnou rezonanční kapacitu na ladicím kondenzátoru. Pro odhad můžeme vyjít z Thompsonova vzorce $f_{RES} = 1/[2 \times \pi \times (LC)^{1/2}]$. S indukčností úzce souvisí ještě činitel jakosti antény, který ovlivňuje šířku pásma. V našem případě potřebujeme alespoň 1,7 MHz, podle čehož upravíme hodnotu sériového odporu antény. Zde vyjdeme ze vztahu $R > 3{,}4 \times 10^6 \times \pi \times L$, přičemž volíme R co nejmenší, abychom anténu netlumili zbytečně.

Na anténu navazuje klasický diodový detektor AM, k němuž je přes odpor šetřící přepěťové ochrany PICu připojen vstup vnitřního komparátoru RA1. Výstup tohoto komparátoru je dále vnitřně (viz zapojení RA4 v [3]) veden na vstup čítače TMR0, který je použit k převodu modifikovaného Millerova kódu na nemodifikovaný. Z jeho lsb si pak data odebírá čtecí smyčka procedury MillerDecoder (viz dále). Propojení s RB3 slouží k řízení záchytného registru pro další vnitřní časovač – TMR1, který je využit k synchronizaci rámců odpovědi podle poslední hrany terminálu (viz norma ISO 14443-3A). Dále je k anténě připojen modulátor tvořený Graetzovým můstkem se zátěží ovládanou tranzistorem MOSFET s indukovaným kanálem N. Použití této technologie je žádoucí, neboť spínací vlastnosti klasického bipolárního tranzistoru řízeného asymetrickým vstupním napětím se ukázaly být nedostatečné pro frekvenci pomocné nosné 847,5 kHz. Připomeňme, že prahové napětí tranzistoru určuje minimální provozní napětí přípravku. Použitý typ vyhovuje napájení od 4,5 V, nižší rozdíl potenciálu si vyžaduje zvážení jiného tranzistoru. Modulační zátěž má převážně odporový charakter, velikost lze regulovat zapojeným trimrem.

Pro ladicí účely a odposlech terminálu je v zapojení i programu počítáno s připojením sériového rozhraní. Připomeňme nutnost úpravy napěťových úrovní podle toho, jaké zařízení a jak chceme připojit. Pro PC využijeme například oblíbený MAX232, případně některý z čipů FTDI určených pro rozhraní USB. My jsme učinili velmi dobrou zkušenost i s modulem Bluetooth OEMSPA311i (www.spezial.cz). Možnost komunikovat s vhodně kamuflovaným PicNicem prostřednictvím technologie Bluetooth samozřejmě velmi usnadňuje experimenty v terénu. Pro snazší ovládání je doplněn obvod automatického resetu při změně úrovně řídicího signálu

DTR, který využívá mj. vnitřní komparátor s řízenou inverzí výstupu.

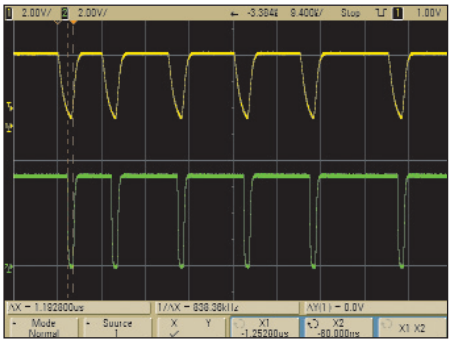
Taktování procesoru je odvozeno z krystalu s frekvencí 13,56 MHz (www.krystaly.cz). Tím je zajištěna možnost solidní synchronizace datových přenosů se čtečkou na úrovni instrukčního toku. Předpokládán je paralelní řez krystalu se zatěžovací kapacitou 30 pF.

Obslužný program

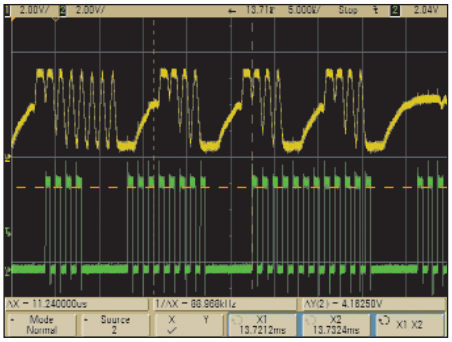
Assemblerový program FakeUID [5] odladěný ve volně dostupném prostředí MPLAB IDE 7.60 (www.microchip.com) umožňuje využít PicNic k podvržení libovolného 4B čísla (UID) karty čtečce, což je v praxi velmi užitečná demonstrační pomůcka. Kromě toho obsahuje obecné prvky, na kterých lze stavět vlastní aplikace. Sem patří zejména procedury MillerDecoder a ManchesterCoder. První slouží k příjmu a dekódování dat ze čtečky s tím, že v této verzi jsou ignorovány paritní bity. Druhá potom umožňuje vyslat zvolený binární řetězec zpět do čtečky. Inicializační kód ošetřuje HW podporu automatického resetu při změně DTR. Pokud není nastaven příznak PCFNoSniff v EEPROM, přejde procesor při DTR = L automaticky do režimu pasivního odposlechu dat terminálu. Ta jsou následně předávána dál přes sériové rozhraní. Při DTR = H je re-aktivován režim emulátoru. Mezi přechody dochází vždy k resetu, který je díky HW podpoře značně nezávislý na momentálním stavu programu. Při nastavení PCFNoSniff zůstává i při DTR = L (po resetu) aktivní modul emulátoru, což umožňuje přes sériové rozhraní sledovat ladicí výstup. Za zmínku dále stojí základní implementace protokolu výběru a stavového automatu karty, která je jednoduchá a zároveň spolehlivě funguje za předpokladu jediné karty v poli. V režimu emulátoru je na úrovni transportní vrstvy implementován speciální příkaz, kterým lze pomocí čtečky (bez nutnosti připojit externí programátor) měnit obsah řadiče EEPROM. Tím lze snadno modifikovat nastavené UID či příznaky pro běh programu. Pro šetření energie zdroje je při delším výpadku pole čtečky aktivován režim spánku procesoru. Další detaily viz zdrojový kód a komentáře v něm.

Oživení

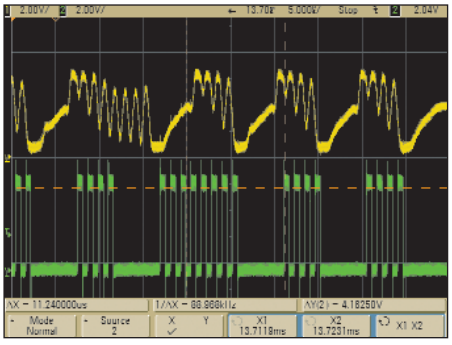
Následující popis předpokládá použití programu FakeUID [5]. Trimr R13 nastavíme na cca 100 Ω. Po zapojení přípravku (4,5–6 V) a nastavení DTR = H přiblížíme anténu ke čtečce, která pouze generuje základní nosnou 13,56 MHz (ověříme magnetickou sondou). Poté nastavíme anténní kondenzátor tak, aby na výstupu detektoru AM (C1, R2) bylo maximální stejnosměrné napětí. Tím jsme naladili anténu, kterou můžeme ještě drobně zapohybovat v poli, abychom si udělali představu o hranicích komunikační vzdálenosti. Aktuální zdrojový kód předpokládá ss napětí od cca 1 V. Dostatečnou sílu pole indikuje LED na RB7 – při pohybu z/do pole by se měla zhášet/rozsvěcet. Nyní zkusíme vyslat příkaz výběru (select). Pokud vše funguje správně, odpoví PicNic číslem, které nám čtečka předá. V opačném případě zkusíme změnu vzdálenosti antény od čtečky v kro-



Obr. 2 Detekce příkazu WUPA (RA1 žlutě, RA4 zeleně)



Obr. 3 Přebuzení čtečky (AUX žlutě, MFOUT zeleně)



Obr. 4 Správná reakce čtečky

ku cca 0,5 cm. Pokud to nepomůže, což by mělo, je na čase provést hlubší diagnostiku. Nejprve je vhodné nastavit DTR = L a zkusit, zda přípravek dekóduje a po sériové lince vysílá správně příkazy čtečky. Pokud tomu tak není, může být na vině příliš vysoká jakost antény, případně chyba v zapojení detektoru a komparátoru. Proměřením obvodu jedno po druhém vyloučíme a případně upravíme sériový odpor antény. Správnou funkci ukazuje *obr. 2*. Pokud dekodér pracuje správně, nastavíme zpět DTR = H a podíváme se na modulátor. V tomto případě je vhodné vyvést ladicí vývod přímo ze čtečky. Vzhledem k tomu, že celý přípravek je koncipován zejména pro experimenty s čipy rodiny Mifare, bude vhodné orientovat se na čtečku vybavenou v tomto prostředí obvyklým komunikačním čipem MF RC531 či novější pinově a instrukčně zpětně kompatibilním CL RC632. Dokumentace [2] je bohužel neveřejná, nicméně lze ji s trochou snahy nalézt na internetu. Konkrétně potřebujeme vyvést piny AUX (27), MFOUT (4) a zem AVSS (28). Číslování odpovídá pouzdru SO32. Dobré zkušenosti máme konkrétně se čtečkou ACR120U (www.rassro.cz). Kromě toho, že uvedený čip je snadno přístupný a šasi čtečky poskytuje dost místa pro konektory, je zásadní výhodou, že programové rozhraní čtečky transparentně zpřístupňuje všechny registry RC531. Možnost jejich zápisu/čtení nabízí dokonce i menu demonstrační aplikace, která je součástí vývojového kitu. Nastavením registru 0x3A na hodnotu 0x04 přivedeme na vývod AUX demodulovanou základní nosnou (13,56 MHz), čili analogový signál pomocné nosné od PicNicu (847,5 kHz), jak ho vidí čtečka (fáze I, hodnota 0x05 je pro fázi Q). Dále nastavíme registr 0x26 na 0x04, čímž na MFOUT získáme obraz pomocné nosné po jejím reformování a digitalizaci. Porovnáním průběhů obou signálů snadno odhalíme chyby v nastavení našeho modulátoru. Průběh na AUX ukazuje, zda čtečka PicNic vůbec „slyší“, zatímco MFOUT ukazuje, zda mu také „rozumí“. Kromě nízké úrovně signálu působí potíže i přebuzení, jehož dopad je vidět na *obr. 3*. Zde čtečka špatně interpretovala sérii krátkých modulačních úseků, které se slily do jednoho. Správnou reakci ukazuje *obr. 4*. Při přebuzení obvykle pomůže oddálení antén, někdy je nutné sáhnout k zatlumení zvýšením hodnoty R13. Je vhodné myslet i na malou šířku pásma antény, která se při příjmu dat nemusela projevit – pak pomůže zvýšení R_ANT.

Závěr

Stavebnice PicNic zpřístupňuje praktické experimenty s emulací čipů RFID v pásmu KV co nejširší komunitě zájemců. Například demonstrovat zranitelnost aplikací založených na pouhém UID je s PicNicem sestaveným na nepájivém kontaktním poli nyní tak snadné…

Vlastimil Klíma, Tomáš Rosa,
v.klima@volny.cz, tomas.rosa@rb.cz

LITERATURA

[1] Lee, Y.: Antenna Circuit Design for RFID Applications, Microchip Tech. Inc., 2003.

[2] MF RC531 – ISO 14443 Reader IC, Philips Semiconductors, Rev. 3.3, 2005.

[3] PIC16F627A/628A/648A Data Sheet, Microchip Tech. Inc., 2007.

[4] Verdult, R.: Security analysis of RFID tags, Master Thesis, 2008.

[5] E-archivy http://cryptography.hyperlink.cz, http://crypto.hyperlink.cz