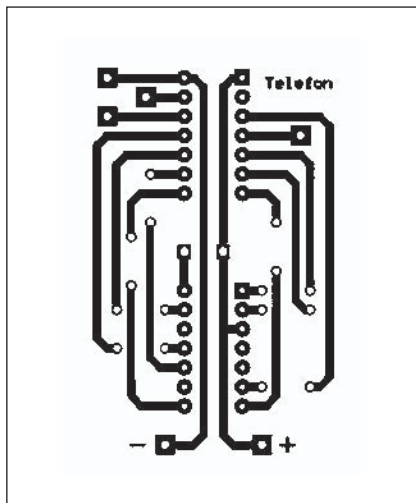
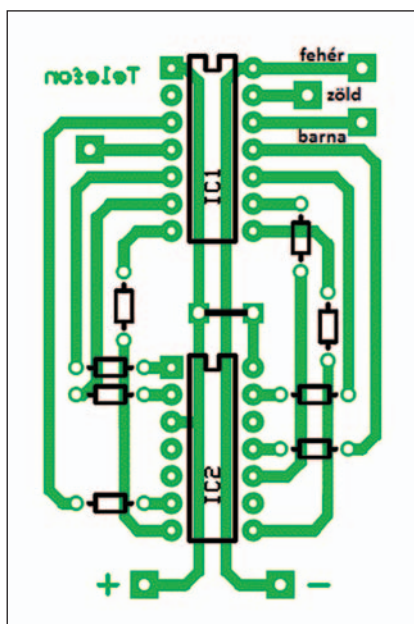


nek is van adattartalma! Továbbá a szinkron adatátvitel idődíagramjain edződött szemünk ahhoz szokott, hogy az adatjel olyan, amilyen, viszont a szinkronjel szinte mindig órajel-szerű. Jelen esetben pedig az **A** jel az, ami órajel-szerű. Igaz, első látásra nem tűnik egyértelműnek az összefüggés a **B** jel hossza és az adat között. Nem is az, mivel a „lényegtelen”-ként jelzett szakasz bármilyen hosszú lehet, mivel az a betekerés ideje, ami az egyéntől függ. Viszont az adatot a „lényeges” szakasz hordozza és a két szakasz pontosan szétválasztható az **A** jel megjelenésével (ez a visszaforgás szakasza). Így azt is mondhatjuk, hogy a **B** adatjel hosszát kell megmérni az **A** szinkronjel megjelenésétől, méghozzá a szinkronjel által diktált időalap szerint. Az érdekes, hogy attól függetlenül, hogy melyik jelet tekintjük adatnak és melyiket szinkronnak, lényegileg mindkét esetben ugyan arra az algoritmusra jutunk. De akkor miből ered ez a dilemma? Abból, hogy mindkét jelnek van adattartalma, azaz a megjelenése, hossza, száma, stb. függ az adat értékétől. Amiatt, hogy az **A** jel kölcsönösen egyértelmű függésben van az adattal, a **B** jel pedig „csak” annyit tesz hozzá az aszinkron átvitelhez, hogy a megjelenésével jelzi az adatátvitel elejét, a megszűnésével pedig a végét, azaz elsődleges a vezérlő funkciója, így én azt mondom, hogy az **A** az adatjel, a **B** pedig a szinkron.



8. ábra



9. ábra

A program Assembly és HEX formátumban a Rádiótechnika honlapjáról letölthető. Előre is elnézést kérek miatta, mivel nem számítok gyakorlottnak eme témában, így a program biztosan nem frappáns és letisztult. Utoljára talán tíz éve kódoztam bele a PIC-ek programozásába, így az elmúlt pár hétvége szabad óráit felhasználva, szinte a nulláról kellett kezdenem a tanulást. Pár gondolatot fűznék a programhoz. Egyrészt végez egy kis animációt a kijelzőn két tárcsázás között, így biztatva a tárcsázásra. A nulla karakter hat szegmense futófény-szerűen körbe-körbe felvillan. Ezt a kijelzett érték körbe-leptetésével érem el. Mivel a kijelző közös anódos, ezért a kimenetek nulla állapotára világítanak a szegmensek, azaz nyelő üzemmódban dolgoznak. Egy másik érdekessége a programnak, hogy szoftveres prell-mentesítést tartalmaz. Ezért nincs a bemeneten kondenzátor. Egy felfutó vagy lefutó él után kb. 40 ms (elég hosszú) ideig a program várakozik, nem vizsgálja a bemenet állapotát. Erre nagy szükség van, mert tapasztaltam, hogy az érintkező nagyon prellés és a processzor bizony képes megszámolni az összes hazárdot. A karakterek bittérképét a program a RAM-ba írja, onnan pedig relatív

címzéssel olvassuk ki. Az előosztó a számlálóhoz van rendelve – maximális osztásaránytal – az időmérés miatt, a watchdogot nem használom.

Az áramkörhöz egyoldalas nyákok is készítettem, melynek rajza a 8. ábrán, beültetése a 9. ábrán, a kész készülék pedig a 10. ábrán látható.

### Miért is?

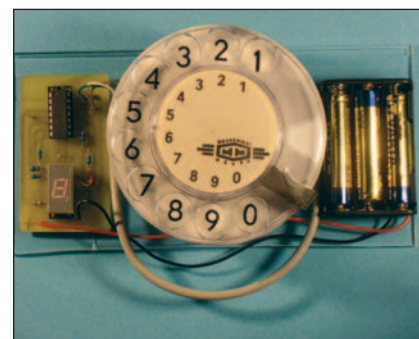
Végezetül egy pár szót arról, hogy miért minimalizáltam a hardvert. Egyrészt ötlet szintjén, egy telefontárcsával, mint bemeneti perifériával és egy PIC-el, ami lehet akár a fenti kistestvére is, a 12F508, mellőzve a kijelzést, olcsón építhetünk egy számkombinációs zárat.

A fő ok azonban a következő volt. A Rádiótechnika több, korábbi számából is értesültem a Ganz Rádióklub fiatalok körében végzett ismeretterjesztő munkásságáról. Tisztelegve ezen missziójuk előtt elhatároztam, hogy az utóbbi változatból összerakok és a klub rendelkezésére bocsátok húsz építőkészletet, amit a gyerekekkel együtt tudnak a következő aktivitásaik alkalmával életre kelteni.

Ehhez kívánok nekik és természetesen az utánépítőknél is sok sikerélményt, jó játékot!

### Források:

1. <http://oldradio.tesla.hu/>
2. <https://www.wikipedia.org/>
3. MAX7219 adatlapja
4. PIC12F508/509/16F505 adatlapja
5. 3LSZ324B1 (3ЛС324Б1) adatlapja
6. Micromite kézikönyv – Dr. Kónya László fordításában



10. ábra