

Név: Schmelczér András

Neptun kód: CVTNXD


Beadás dátuma: 2020. 05. 19.

Házi feladat címe: Szimulációs játék

## Mikrokontrollerek alkalmazástechnikája házi feladat

ATtiny85-ös mikrokontrollerrel szeretnék megvalósítani egy infrás távirányítóval irányítható játékot. A játék futásához nem kell emberi beavatkozás, magától is történnek benne események, viszont, ha fogunk egy távirányítót, akkor az egyik karaktert (a három közül) irányíthatjuk vele. A videójáték űrhajós tematikájú, amiben fejlődni az aszteroidák szétlövésével lehetséges.

A feladatot önállóan, meg nem engedett segítség nélkül oldottam meg:



.....  
aláírás

## Részletes specifikáció

A játék futásához nem kell emberi beavatkozás, magától is történnek benne események, viszont, amennyiben fogunk egy távirányítót, akkor az egyik karaktert irányíthatjuk vele. A rendszer 3.3 V-ról működtethető, fogyasztása nem haladja meg a 15 mW-ot.

Infrán keresztüli NEC protokollal történő kommunikációval hét különböző kérést küldhetünk. A kijelző fényereje állítható növekvő és csökkenő irányban, a képernyőn látható egyik karakter mozgatható négy irányban, illetve a képernyőn látható karakter készíthető cselekvésre.

A rendszerrel továbbá UART-on keresztül is lehet kommunikálni. Küldeni a távirányító által küldhető parancsok karakteres megfelelőjét lehet. Ezek az előző sorrenddel megegyezően a következők: '+', '-', 'w', 'a', 's', 'd', '.'. Visszafele pedig egy képkocka kirajzolásának ideje érkezik pár másodperces időközönként.

Maga a játék az űrben játszódik. Ezen belül egy űrhajóban található a játékos által irányított karakter és pár másik, a gép által irányított karakter is. A karakterek az űrhajót irányíthatják (ehhez az irányítópulthoz kell menni, és megnyomni az akciógombot). Az űrhajóval lőni is lehet, ehhez egy másik irányítópulthoz kell menni. Az űrhajó egy részének a képernyőn mindig látszania kell, így azt nem lehet bármilyen távolra elirányítani.

Vizuálisan oldalnézetű képeket látunk. Az űrhajó kétszintes, ezek között, illetve ezeken belül tudnak a karakterek közlekedni. A két szintet egy létra köti össze, amelyen fel és lefele is lehet közlekedni. A lefele mozgásnak a létrán kívül nincs következménye. A felfele mozgást a létrán kívül végrehajtva egy ugró animációt láthatunk.

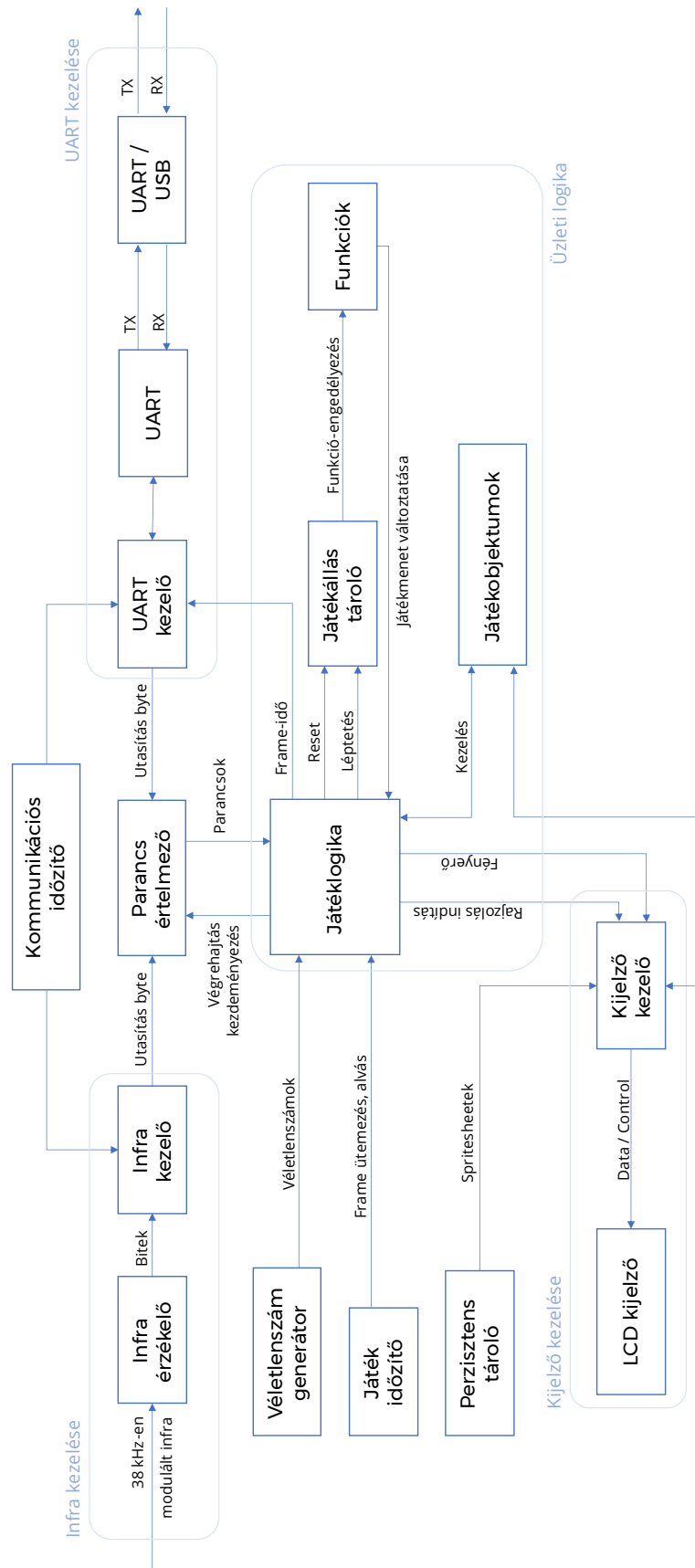
Ütközésnek számít az és csak az, amikor az űrhajó legalább egy sötét pixele fedésben van az aszteroida egy sötét pixelével. Az űrhajó ütközhet az aszteroidákkal, ilyenkor az energiája csökken. Az energia lövés hatására is csökken. Erről visszajelzést egy az űrhajó spritejába integrált csík szolgáltat. Ha a csík hosszúsága nulla, a játéknak vége, és egy effekt lejátszása után előlről kezdődik. Ezt elkerülendő a fedélzet megfelelő pontján kell az akciógombot megnyomni, ilyenkor újra feltöltődik a csík. Néha megjelennek aszteroidák, melyek szétlőhetők.

A spriteok animálhatók és az átlátszóság is biztosított. A háttérben csillagok mozognak véletlenszerű konfigurációkban. Utóbbiaknak nincs hatása a játékmenetre. Követelmény továbbá, hogy legalább 50 képkockát kirajzoljon másodpercenként legalább 64 x 32 pixeles felbontásban.

A játéknak figyelembe kell vennie az eddig felrobbantott aszteroidák számát, és ennek függvényében újabb funkciókat kell a játékos számára biztosítani. Ezek a funkciók sorrendben a következők: energia növelés lehetősége, lövés lehetősége, első segítő végül pedig a második segítő.

A segítők másik karaktereket jelölnek, akik egy többszintes szabályrendszert követve próbálják a játék előrehaladását segíteni. Minden funkció elérhető számukra, ami a játékos rendelkezésére is áll.

# Funkcionális blokkvázlat



## Részletesebben

A specifikáció alapján az előző oldalon található blokkvázlatot készítettem. Sokat gondolkoztam a megfelelő absztrakciós szinten. Végül amellettt döntöttem, hogy a játéklógika fáradságos részletezése helyett annak csak a specifikációban kiemelt fő részeit tüntetem fel. Az eszköz viselkedésére már ennyiből is kellően nagy pontossággal lehet következtetni, a pontos implementáció és felbontás pedig rendkívül sokféle lehet. Szerencsére a többi követelmény elég egyértelműen vetíthető funkcionális blokkokra.

Mivel parancsokat több forrásból is kaphat a rendszer így a komplexitás csökkentése végett, azokhoz közös feldolgozóegységet célszerű használni. A kommunikációs protollok bonyolultsága miatt ezekhez kezelőket is célszerű alkalmazni, így a rendszer többi része könnyebben veheti igénybe őket. Ezeknek a kezelőknek időzítőre is szükségük lesz.

Többször említésre került a követelmények közt a véletlenszerűsége igénye, ezek megvalósításához szükséges egy véletlenszám-generátor is. Továbbá explicit nem szerepelt a perzisztens tároló szükségessége, viszont az animált sprite-ok tárolását ezzel megoldhatjuk.

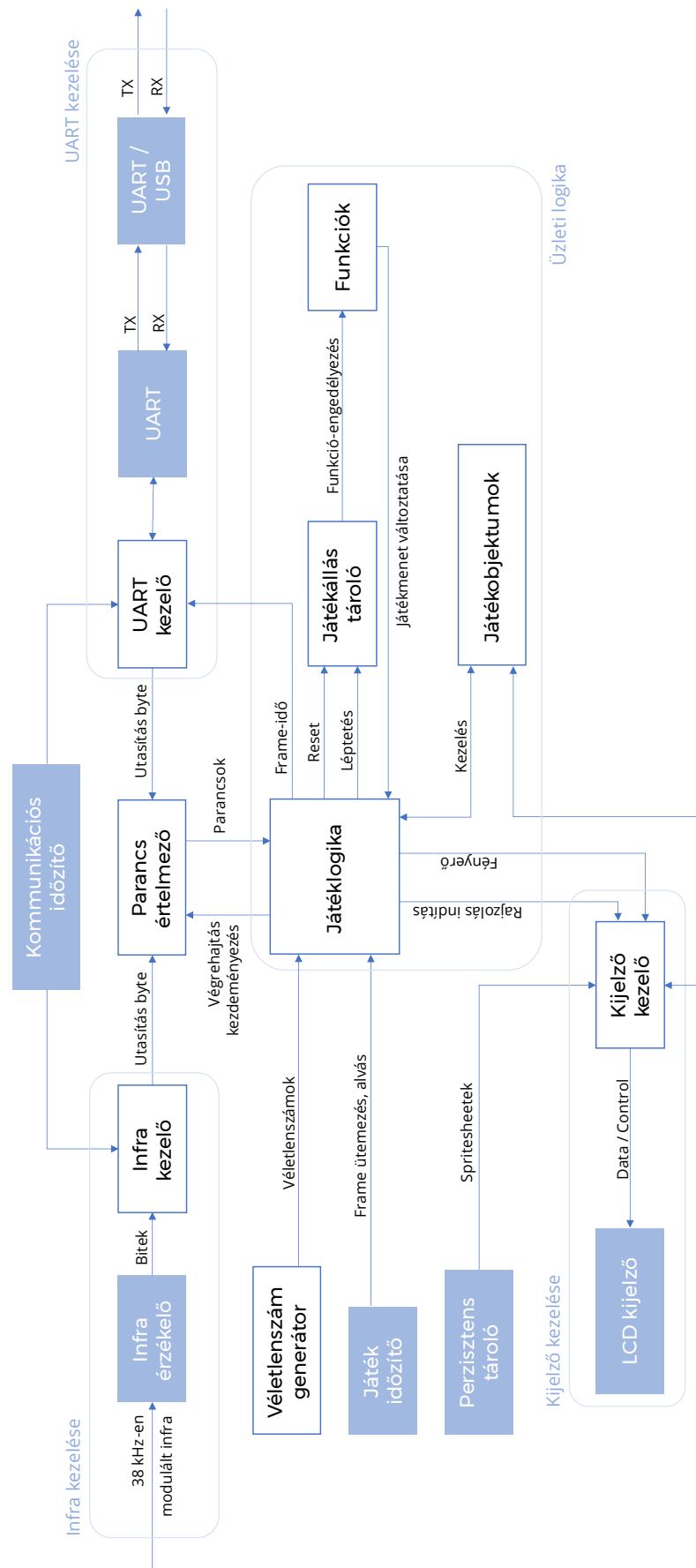
Az üzleti logika felel a követelményrendszer nagyobb részének teljesítéséért. Ennek kiemelendő funkciói a játékállapotának és a játék objektumoknak a felügyelete. Emellett a többi komponens működésének összeegyeztetése is ezen egység feladata.

## Hardver szoftver szétválasztás

Ezek után a hardver szoftver szétválasztás többnyire egyértelmű lesz. A ki és bemenetek (infra, UART, LCD) természetesen hardveresen kerülnek megvalósításra. Az időzítők és tárhely pedig a mikrokontrollerek felépítéséből következően szintén hardveresen manifesztálódnak. Jó hír, hogy ezeken kívül minden funkció megvalósítható szoftveresen is. A felbontást vizuálisan a következő oldalon láthatjuk.

## Diagram

A hardver által megvalósított részeket kék háttérszínű téglalapokkal jelöltem.



## Hardver blokkvázlat

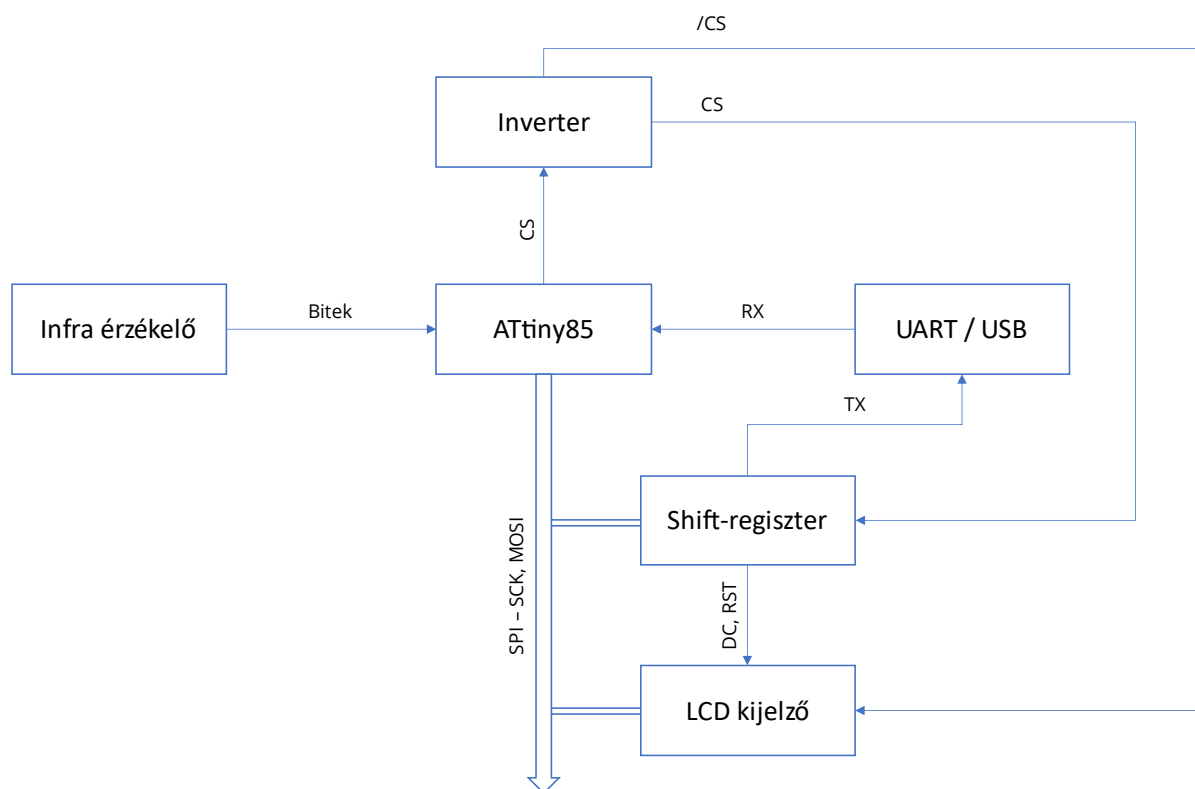
Maga a mikrokontroller a korábban említett ATtiny85-ös lesz. Ez egy kellően olcsó és kisméretű eszköz, meglepően magas teljesítménnyel. Hátránya a kisméretű RAM és programmemória. Ezeket ügyes programozással kell áthidalni.

Egy 128 x 64 pixel felbontású SPI-on kommunikáló kijelző szolgál kimenetként, egy TSOP szériás infra érzékelő pedig bemenetként. A kijelző felbontása (és beépített hardveres gyorsítása) lehetővé teszi a kétszeres upscaling használatát, így elég 64 x 32 pixelnyi képet kiszámolni. Fejlesztés alatt a mikrokontroller debugWIRE interface-én kerül sor a debugolásra. A tárgykövetelmények miatt UART-os kommunikációra is szükség lesz, viszont a lábak elfogytak, így a kimeneti lábakat egy shift-regiszterrel kell bővíteni, amivel kvázi SPI-on keresztül lehet kommunikálni.

A (kimeneti) portbővítés az eddigi funkcionális leírásokban nem szerepelt, hiszen ez nem a specifikációban részletezett funkcionalitáshoz szükséges, hanem egyszerűen a mikrokontroller kis lábszáma által okozott akadályt szünteti meg.

A két SPI lábakat használó periféria miatt szükséges a chip-select jelek használata, ezt egy inverter alkalmazásával biztosítom.

A mikrokontroller beépített USI-ja az SPI kezelésre lesz konfigurálva, lévén ez az időkritikusabb a relatíve nagy sávszélességet igénylő képkockák átküldése miatt. Az UART és az infra kezelésére pedig maradnak a pin change és timer interruptok.



[illegible]

## Szoftver rendszerterv

A teljes szoftveres dokumentációt mérete miatt nem illesztettem be ebbe a dokumentációba. A kommentek és a kód felépítése alapján doxygen-nel és pdlatex-kel generált dokumentáció a `software_documentation.pdf` fájlban található meg nyomtatható verzióban. Böngésző segítségével célszerűbb a kereshető `software_documentation/index.html` fájljon keresztül elérhető dokumentációt használni.

Ebben a részben csak átfogóbb képet szeretnék adni a rendszerről. A részleteket az előbb említett helyen lehet megtalálni.

### Elvek

A szoftver architektúrájának tervezésekor fő szempontok voltak a modularitás, hordozhatóság és a teljesítmény. Nem elhanyagolható igény továbbá a kis méret sem. Mindezek teljesítése végett több alkalommal ismerhető fel az objektum-orientált szemlélet.

Ezek miatt a programban nincs globálisan elérhető állapot, a komponensek saját állapotukat felügyelik, velük függvényhívásokkal lehet kommunikálni.

### Hardverközele

Madártávlatból három fő részre osztható a kódbázis. Legalul helyezkedik el a hardware-access, ami a legalapvetőbb hardveres funkciókat látja el egy kényelmesen használható API-val. A következő hardver funkciókért felelős modulok találhatók itt.

- Energiahatékonyság
- SPI (sebesség végett blokkoló működés)
- Időzítés (megszakítás alapú működés)
- Digitális kimenet

Másik mikrokontrollerre váltás esetén a változtatások túlnyomó része az előbbi elemekben várható.

Ez a réteg rendelkezik egy Façade jellegű interfésszel, aminek két felelőssége van. Egyrészt deklarálja a hardware-access által biztosított funkciókat. Másrészt a teljes hardveres inicializálásért felel. Utóbbira a kis méret és az egyszerű használat miatt van szükség. A program elején inicializáljuk a HAL-t, utána pedig bárhol includeolva a megfelelő headert, a teljes hardver funkcionalitást elérjük.

### Driverrek

A következő réteg a driver réteg. Az itteni modulok lazán csatoltak és egyetlen közös tulajdonságuk, hogy a HAL-t használják funkcionalitásuk megvalósításához.



A következő komponensek helyezkednek el itt.

- Kijelző
- Infra (megszakítás alapú működés)
- Alvásütemezés
- UART (megszakítás alapú működés)

Ezek többnyire egy egyszerű interfész (callback függvény) átadásával inicializálhatók. Erre a laza csatoltság miatt van szükség, igazából ez a dependency inversion principle C nyelvi megvalósítása.

A legmagasabb szintű funkciók az alkalmazás réteg-ben található. Maga a játék és annak logikája itt kerül megvalósításra. Ez saját belső architektúrával rendelkezik, amit a későbbiekben fogok kifejteni.

Végül egy fontos komponens maradt még. A *mediator* az alkalmazás és a driverek között helyezkedik el. Biztosítja, hogy az előbbi két rétegen belül és azok közt megvalósulhasson a kommunikáció. Felelőssége a komponensek orkesztrációja. Mindezt a megfelelő fejlécű függvények átadásával éri el a megfelelő inicializáló függvényhívásokban.

## Alkalmazáslogika

A fő szempont a magasszintű kód írásának lehetővé tétele és a kompakt méret. Ezért a játékobjektumok közt egy prototípus alapú öröklési rendszer került kialakításra. Minden típus rendelkezik egy prototípussal, ami a méretét, a rajzoló és a frissítő függvényét tartalmazza. Maguk a prototípusok a programmemóriában kerülnek tárolásra. Az objektumok rendelkeznek egy mutatóval a prototípusukra, egy pozícióval és egy union-nal, amiben a típus specifikus állapotukat tárolják.

Az objektumok tárolására az objektumtároló szolgál. Ez igazából egy heterogén kollekció pár hasznos függvénnyel. Például segítségével lekérdezhetők az objektumok típus alapján vagy az alapján, hogy adott pixel pozíciót tartalmazzák-e. Továbbá az objektumok frissítését és rajzolását lehet innen kezdeményezni.

Az időközönkénti új objektumok létrehozását az esemény generátor kezdeményezi. Emellett a játékmenetért (indítás, újraindításért) felelős komponens is ebben a rétegben kerül megvalósításra.

Az összes játékobjektum típus rendelkezik saját komponenssel. Adott feladatukat a program többi részét használva igen magas szintű kóddal lehet megvalósítani. Önálló komponense van továbbá az AI-nak, ami az élő felhasználó által használható interfészen keresztül irányíthatja a karaktereket.

Megemlíthető, hogy a kirajzolásra szánt spritesheetek is itt kerülnek definiálásra a kijelző által elvárt speciális formátumban. Ezek az EEPROM-ban kerülnek tárolásra.

## Segédfüggvények

Egyéb sztenderd könyvtár jellegű segédfunkciók is helyet kapnak a kódbázisban. Ezek működése elég magától értetődő. Felsorolás jelleggel a következő komponensekről van szó.

- Véletlenszámgenerátor
- Vektor (2 dimenziós)
- Téglalap
- Makrók
  - Null
  - Bit műveletek

## Felhasználói leírás és kezelési útmutató

Az eszköz bekapcsolásához szükséges egy 3.3 V-os tápfeszültséget biztosítani számára. Ezután az UART – USB átalakító csatlakoztatva a számítógéphez, egy terminál alkalmazással (például PuTTY) csatlakozhatunk az eszközhöz.

Ilyenkor az OLED kijelzőn egy karaktert láttunk az űrhajóban állni. Ezt a karaktert kétféleképpen irányíthatjuk. Egyrészt használhatjuk az imént elindított soros kommunikációt, vagy az eszközhöz tartozó infrás távirányítót is. Előbbi esetén a 'w', 'a', 's', 'd' és szóköz billentyűk lenyomásával mozgathatjuk a karaktert, illetve vehetjük rá, hogy interakcióba lépjen a környezetével. A '+' és '-' billentyűk lenyomásával a kijelző fényerejét változtathatjuk. Távirányító esetében ezeknek a '2', '4', '8', '6', '5', '+' és '-' billentyűk felelnek meg. Bizonyos időközönként a soros porton egy 'szám ms' alakú üzenet érkezik, ami megadja, hogy az előző ilyen üzenet óta mennyi időbe telt a leglassabban előállított kép kirajzolása.

A játék célja, az aszteroidák felrobbantása, miközben az űrhajó épséget megőrizzük. Aszteroidák felrobbanthatóak, ha az űrhajóval ütköznek, vagy lelövik őket. Az űrhajó mozgatásához annak jobb felső részéhez érve kell interakció gombot nyomni. Ettől kezdve a karakter helyett az űrhajót mozgatják a gombnyomások. Visszaváltani az interakció gomb ismételt lenyomásával lehetséges.

Kellő mennyiségű felrobbantott aszteroida után az űrhajó felszereltsége nő. Az újonnan megjelenő tárgyakkal is interakcióba lehet lépni. Bal alul az űrhajót javíthatjuk (a jelenlegi állapotát jobb középen elhelyezett csíkon láthatjuk). Jobb alul található az ágyú. A felszereltség növekedése mellett gép által irányított űrhajósók is megjelenhetnek, akik elvárható pontossággal végzik a szükséges feladatokat a játék célja eléréseért.

Ha a korábban említett csík hossza nullára csökken, egy animáció lejátszása után a játék előlről kezdődik.

## Bemérési tapasztalatok

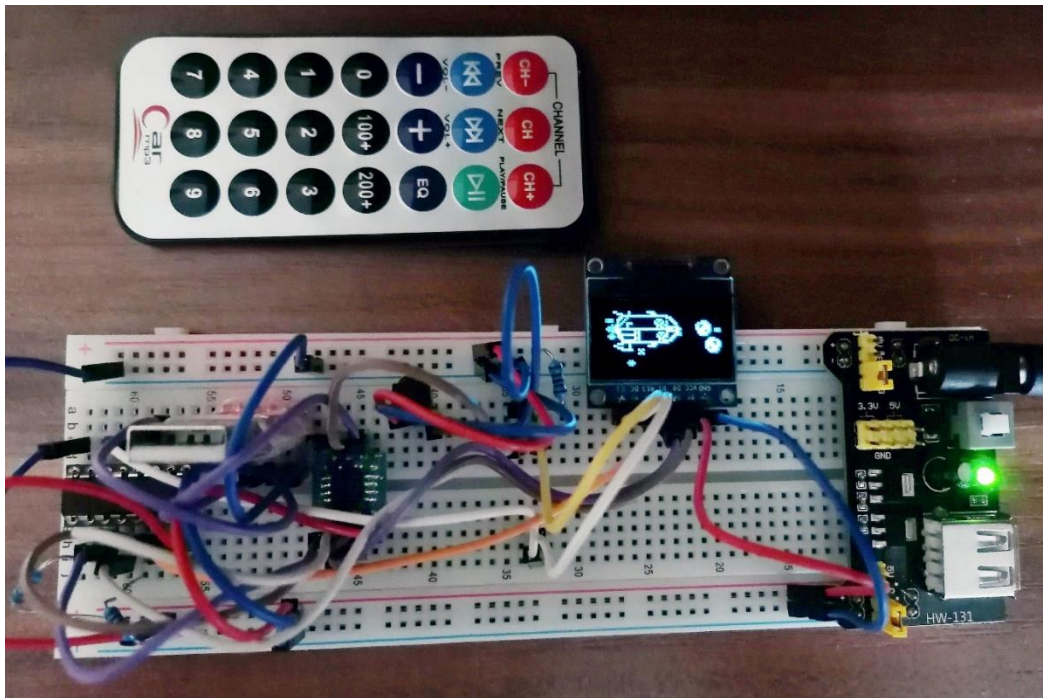
Többször volt szükséges átépíteni a hardvert amiatt, mert az olcsó és kisméretű breadboard-ok csatlakozói nagyon hamar elhasználódnak. Emiatt legközelebb mindenképpen egy komolyabb breadboard-on kezdeném el a prototipizálást.

Emellett a fejlesztés első fázisában fejlesztőkártyát (vagy külső debugger-t) használnék a beépített debugolási lehetőségek miatt. Továbbá célszerű, ha mindenből legalább egy tartalék van nálunk, erre nemegyszer volt szükségem.

Alkatrészek kapcsán legpozitívabban a TSOP-ban csalódtam, nagyon jólszűrt és erős jelet ad még akkor is, ha az infra forrása relatíve messze van tőle.

## Elkészült eszköz

Egy kép a dokumentációban részletezett eszköz breadboardon elkészült prototípusáról.



Az előző eszköz alapján készített, továbbfejlesztett konfiguráció pedig ezen a képen látható. Utóbbiról több információ a <https://schmelczér.dev> oldalon található.



## Datasheet borítók

A következőkben a felhasznált hardverek datasheetjeinek első oldalát másoltam be. Ezek a következő chippekhez tartozó borítók:

- 2N7000G
- 74HC595
- D096128-SPI-7
- ATTINY85-V
- TSOP48

### Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 120 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
- Non-volatile Program and Data Memories
  - 2/4/8K Byte of In-System Programmable Program Memory Flash (ATtiny25/45/85)  
Endurance: 10,000 Write/Erase Cycles
  - 128/256/512 Bytes In-System Programmable EEPROM (ATtiny25/45/85)  
Endurance: 100,000 Write/Erase Cycles
  - 128/256/512 Bytes Internal SRAM (ATtiny25/45/85)
  - Programming Lock for Self-Programming Flash Program and EEPROM Data Security
- Peripheral Features
  - 8-bit Timer/Counter with Prescaler and Two PWM Channels
  - 8-bit High Speed Timer/Counter with Separate Prescaler  
2 High Frequency PWM Outputs with Separate Output Compare Registers  
Programmable Dead Time Generator
  - USI – Universal Serial Interface with Start Condition Detector
  - 10-bit ADC
    - 4 Single Ended Channels
    - 2 Differential ADC Channel Pairs with Programmable Gain (1x, 20x)
    - Temperature Measurement
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - debugWIRE On-chip Debug System
  - In-System Programmable via SPI Port
  - External and Internal Interrupt Sources
  - Low Power Idle, ADC Noise Reduction, and Power-down Modes
  - Enhanced Power-on Reset Circuit
  - Programmable Brown-out Detection Circuit
  - Internal Calibrated Oscillator
- I/O and Packages
  - Six Programmable I/O Lines
  - 8-pin PDIP, 8-pin SOIC and 20-pad QFN/MLF
- Operating Voltage
  - 1.8 - 5.5V for ATtiny25/45/85V
  - 2.7 - 5.5V for ATtiny25/45/85
- Speed Grade
  - ATtiny25/45/85V: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
  - ATtiny25/45/85: 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Industrial Temperature Range
- Low Power Consumption
  - Active Mode:
    - 1 MHz, 1.8V: 300 µA
  - Power-down Mode:
    - 0.1 µA at 1.8V



**8-bit AVR®  
Microcontroller  
with 2/4/8K  
Bytes In-System  
Programmable  
Flash**

**ATtiny25/V\*  
ATtiny45/V  
ATtiny85/V\***

**\*Preliminary**

2586J-AVR-12/06



## 2N7000G

### Small Signal MOSFET 200 mAmps, 60 Volts N-Channel TO-92

#### Features

- ☐ AEC Qualified
- ☐ PPAP Capable
- ☐ This is a Pb-Free Device\*

#### MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Drain Source Voltage	$V_{DS}$	60	Vdc
Drain-Gate Voltage (RGS = 1.0 MW)	$V_{DGR}$	60	Vdc
Gate-Source Voltage	$V_{GS}$	$\pm 20$	Vdc
- Continuous	$V_{GSM}$	$\pm 40$	Vpk
- Non-repetitive ( $t_p \leq 50$ ms)			
Drain Current	$I_D$	200	mA dc
- Continuous	$I_{DM}$	500	
- Pulsed			
Total Power Dissipation @ $T_C = 25^\circ\text{C}$	$P_D$	350	mW
Derate above $25^\circ\text{C}$		2.8	mW/ $^\circ\text{C}$
Operating and Storage Temperature Range	$T_J, T_{stg}$	-55 to +150	$^\circ\text{C}$

#### THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction-to-Ambient	$R_{qJA}$	357	$^\circ\text{C/W}$
Maximum Lead Temperature for Soldering Purposes, 1/16" from case for 10 seconds	$T_L$	300	$^\circ\text{C}$

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

\*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.



**ON Semiconductor®**

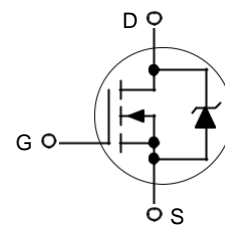
<http://onsemi.com>

**200 mAmps**

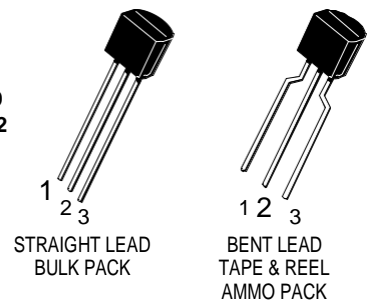
**60 Volts**

**$R_{DS(on)} = 5\text{ W}$**

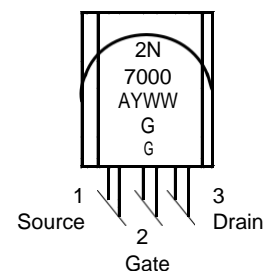
**N-Channel**



**TO-92  
CASE 29  
STYLE 22**



#### MARKING DIAGRAM AND PIN ASSIGNMENT



A = Assembly Location  
Y = Year  
WW = Work Week  
G = Pb-Free Package  
(Note: Microdot may be in either location)

#### ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 2 of this data sheet.

**INTEGRATED CIRCUITS**

# DATA SHEET

## **74HC595; 74HCT595**

8-bit serial-in, serial or parallel-out shift register with output latches; 3-state

Product specification  
Supersedes data of 1998 Jun 04

2003 Jun 25

**SOLOMON SYSTECH**  
**SEMICONDUCTOR TECHNICAL DATA**

---

**SSD1306**

***ADVANCE INFORMATION***

**128 x 64 Dot Matrix**  
**OLED/PLED Segment/Common Driver with Controller**

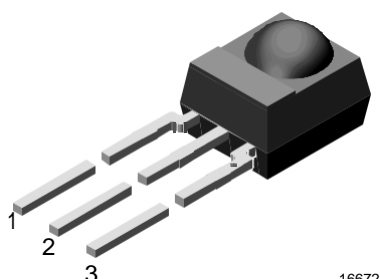



[www.vishay.com](http://www.vishay.com)

# TSOP22..., TSOP24..., TSOP48..., TSOP44...

Vishay Semiconductors

## IR Receiver Modules for Remote Control Systems



16672

### MECHANICAL DATA

#### Pinning for TSOP44..., TSOP48...:

1=OUT, 2=GND, 3=VS Pinning

#### for TSOP22..., TSOP24...:

1=OUT, 2=VS, 3=GND

### FEATURES

- Improved immunity against HF and RF noise
- Low supply current
- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Supply voltage: 2.5 V to 5.5 V
- Improved immunity against optical noise
- Insensitive to supply voltage ripple and noise
- Material categorization:  
for definitions of compliance please see  
[www.vishay.com/doc?99912](http://www.vishay.com/doc?99912)



**RoHS**  
COMPLIANT  
HALOGEN  
**FREE**  
**GREEN**  
(5-2008)

### DESCRIPTION

The TSOP22..., TSOP48..., TSOP24.. and TSOP44.. series are miniaturized IR receiver modules for infrared remote control systems. A PIN diode and a preamplifier are assembled on lead frame, the epoxy package contains an IR filter.

The demodulated output signal can be directly connected to a microprocessor for decoding.

The TSOP24..., TSOP44.. series devices are optimized to suppress almost all spurious pulses from Wi-Fi and CFL sources. They may suppress some data signals if continuously transmitted.

The TSOP22..., TSOP48.. series devices are provided designs should prefer the TSOP24..., TSOP44.. series containing the newer AGC4.

These components have not been qualified according to automotive specifications.

### PARTS TABLE

AGC		LEGACY, FOR LONG BURST REMOTE CONTROLS (AGC2)		RECOMMENDED FOR LONG BURST CODES (AGC4)	
Carrier frequency	30 kHz	TSOP4830	TSOP2230	TSOP4430	TSOP2430
	33 kHz	TSOP4833	TSOP2233	TSOP4433	TSOP2433
	36 kHz	TSOP4836	TSOP2236	TSOP4436 (1)(2)(3)	TSOP2436 (1)(2)(3)
	38 kHz	TSOP4838	TSOP2238	TSOP4438 (4)(5)(6)	TSOP2438 (4)(5)(6)
	40 kHz	TSOP4840	TSOP2240	TSOP4440	TSOP2440
	56 kHz	TSOP4856	TSOP2256	TSOP4456 (6)(7)	TSOP2456 (6)(7)
Package		Mold			
Pinning		1=OUT, 2=GND, 3=VS	1=OUT, 2=VS, 3=GND	1=OUT, 2=GND, 3=VS	1=OUT, 2=VS, 3=GND
Dimensions (mm)		6.0 W x 6.95 H x 5.6 D			
Mounting		Leaded			
Application		Remote control			
Best remote control code		(1) RC-5 (2) RC-6 (3) Panasonic (4) NEC (5) Sharp (6) r-step (7) Thomson RCA			