

Arduino autokassza Fejlesztői dokumentáció

Készítette: Pelz Ádám Márk

Tartalomjegyzék

1. Specifikációk	3
1.1 Szoftver specifikációk	3
1.2 Hardver specifikációk	3
1.3 Az eszköz tervrajza	5
2. Komponensek leírása	5
2.1 HX711 Scale Load Cell	5
2.2 MH-ET LIVE Scanner v3.0	6
2.3 W5500 Ethernet Module	6
2.4 16x2 LCD kijelző	6

1. Specifikációk

1.1 Szoftver specifikációk

- SQL alapú adatbázisszoftver (amit én használtam: MariaDB)
- Webszerver alkalmazás (amit én használtam: Apache2)
- Arduino IDE

1.2 Hardver specifikációk

A projekt összerakásához az alábbi komponensek szükségesek.

HX711 Scale Load Cell 1 db



MH-ET LIVE Scanner v3.0 1 db



W5500 Ethernet Module 1 db



16x2 LCD kijelző 1 db



Potenciométer 1 db



Zöld színű LED dióda 1 db



Piros színű LED dióda 1 db



Gomb 1 db



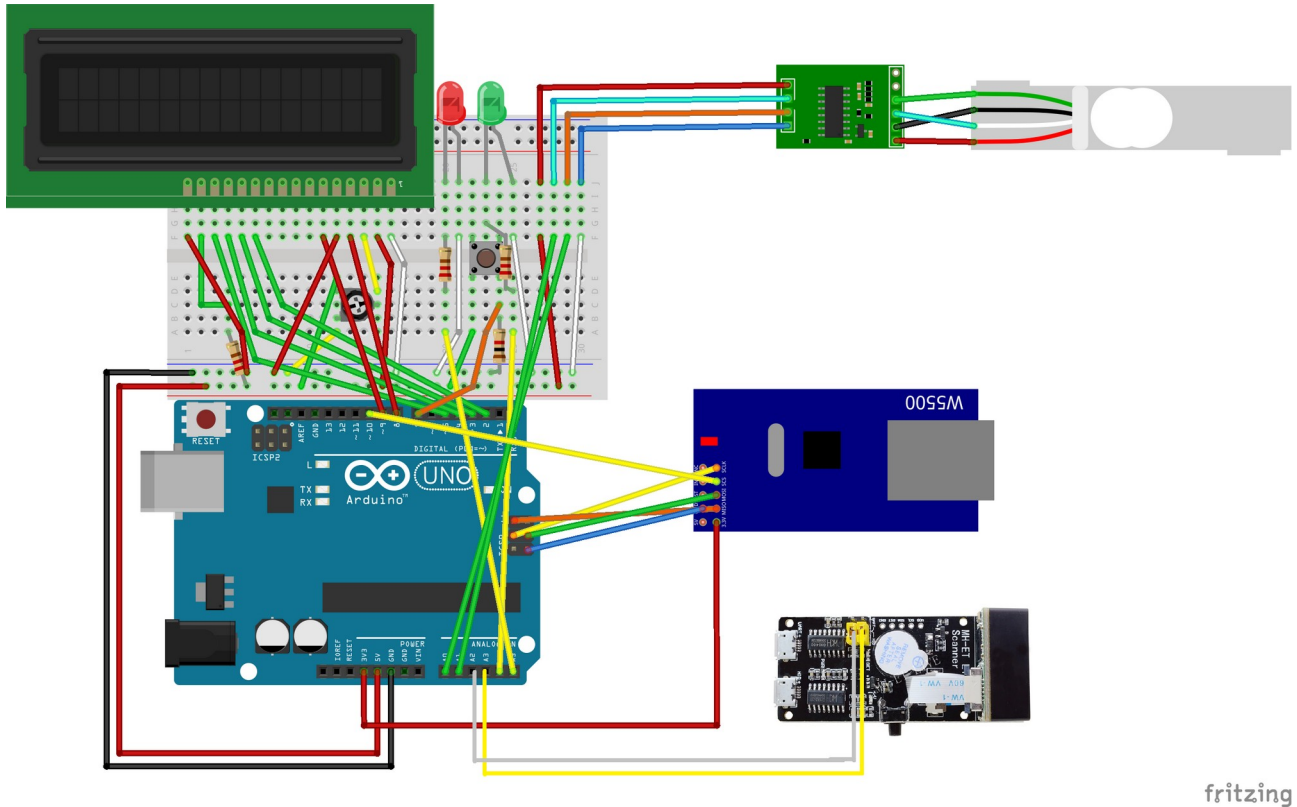
220Ω ellenállás 3 db



1kΩ ellenállás 1 db



1.3 Az eszköz tervrajza



2. Komponensek leírása

2.1 HX711 Scale Load Cell

Az HX711 Scale Load Cell segítségével történik az áruk és az érmék súlyának mérése. A HX711 könyvtár

A HX711 könyvtár függvényei:

- `begin()`: Pinek inicializálására.
- `set_scale()`: Mérleg kalibrálására.
- `tare()`: Mérleg kinullázása.
- `get_units()`: Adatok kinyerése.

A program futása közben a mérleg által mért értéket kell átalakítani gramm mértékegységre, ehhez viszont kalibrálni kell a mérleget. A kalibrálás során a mérleg által mért érték és egy ismert súly alapján fogunk egy kalibrálási értéket kiszámítani.

Először meg kell hívni a `set_scale()` függvényt paraméter nélkül, majd a `tare()` függvényt is paraméter nélkül. Ezután a mérlegre kell helyezni a teszt súlyt és a `get_units(10)` függvénnyel kiolvasni az értéket, amit mér. A kiolvasott értéket elosztva a teszt súly valós súlyával megkapjuk a form faktort, amit tovább kell adni a `set_scale()` függvénynek.

A kalibrálás elvégezte után a mérleg folyamatosan mérési adatokat fog küldeni a gép számára a `get_units()` függvény meghívásával.

2.2 MH-ET LIVE Scanner v3.0

Vonalkódok olvasására szolgál, amik alapján be lehet egy terméket azonosítani. A SoftwareSerial könyvtárat használja, ami nem csak ennél az eszköznél használható.

A SoftwareSerial könyvtár függvényei:

- `begin()`: Eszköz inicializálása paraméterként megadott sebesség alapján.
- `available()`: Ellenőrzi, hogy van-e kiolvasható karakter.
- `read()`: Egy karakter kiolvasása.

A program futása közben folyamatosan ellenőrizni fogja, hogy van-e a scannerből kiolvasható adat. Az `available()` függvény adja meg mennyi karakter vár éppen beolvasásra, ennek jelen esetben hatnak kéne, lennie, mivel korábban ennyi lett definiálva, de mégis nyolcat várunk, hiszen egy null terminátort rak a karaktertömb végére. Ha ez megvan, akkor a beolvasott karakterek egyenként eltárolódnak, és összehasonlításra kerülnek a rendszerben lévő vonalkódokkal. Az összehasonlítás egy saját `comparison()` függvényben történik, ami egy bool értéket ad vissza.

2.3 W5500 Ethernet Module

Hálózati kapcsolat megvalósításáért felel. Szükségesek hozzá az SPI és az Ethernet könyvtárak.

Az Ethernet könyvtár függvényei:

- `init()`: Egyetlen egy pint kell inicializálni vele, jelen esetben a D10-est.
- `begin()`: Hálózati működéshez szükséges adatokat lehet megadni, pl. MAC cím, IP cím, átjáró, alhálózat

Az eszköz beállítása után nincs vele dolgunk. A projekthez létre kell hozni egy `EthernetClient` objektumot.

2.4 16x2 LCD kijelző

A vásárlás jelenlegi állapotát fogja közölni a vásárlóval, kisebb utasításokat fog kiadni. A LiquidCrystal könyvtáron keresztül lehet működtetni.

A LiquidCrystal könyvtár függvényei:

- `begin()`: Pinek inicializálása.
- `setCursor()`: Pozíció beállítása.
- `print()`: Karakterek írása.
- `clear()`: Képernyőtörlés.

Minden egyes alkalommal, ha írunk érdemes egy képernyőtörlést végezni, majd a kurzor állításával lehet megkezdeni az írást.