

Óbudai Egyetem



# RFID ÓRA IDŐMÉRÉS LABVIEW ALKALMAZÁSSÁVAL

projektum  
Intelligens Rendszerek tárgyból

témavezető: Dr.Simon János  
főiskolai tanár

hallgató: Kovács Árpád  
Neptun kód: BPJZ56

Szabadka, 2020

# Tartalom

Bevezető.....	3
1. Projektfeladat .....	3
2. Elméleti alapok.....	3
MySQL adatbázis:.....	8
3. A LabVIEW szoftver bemutatása.....	9
A felhasznált rövidítések .....	16
Irodalom.....	16

# Bevezető

A dokumentum az intelligens rendszerek tárgy dokumentálásának céljából jött létre.

## 1. Projektfeladat

Az én projekt feladatom az volt, hogy egy adatbázis segítségével mérjem az időt az rfid kártyák lecsippantásával. Ehhez a feladathoz ESP32 mikrovezérlőt használtam fel.

## 2. Elméleti alapok

Labview Program:

A labview az NI által fejlesztett ún. Virtuális műszerek létrehozására, és grafikus programozásra használt szoftver.

ESP32 egy 32 Mb os flash kapacitású Wifi-vel és Bluetooth-al rendelkező MCU.

Kisméretű modulok családja, amelyek ESP32 chipet és számos több kulcsfontosságú komponenst tartalmaznak, beleértve a kristály oszcillátort és MIFA antennát. Ez megkönnyíti az ESP32 alapú eszközök használatát, amely így készen áll a végtermékekbe való beszerelésre. Ezek a modulok fejlesztési platformként is használhatók, mivel számos egyéb összetevőt is tartalmaznak, például programozási felületet, a passzív alkatrészeket és túsorosot.

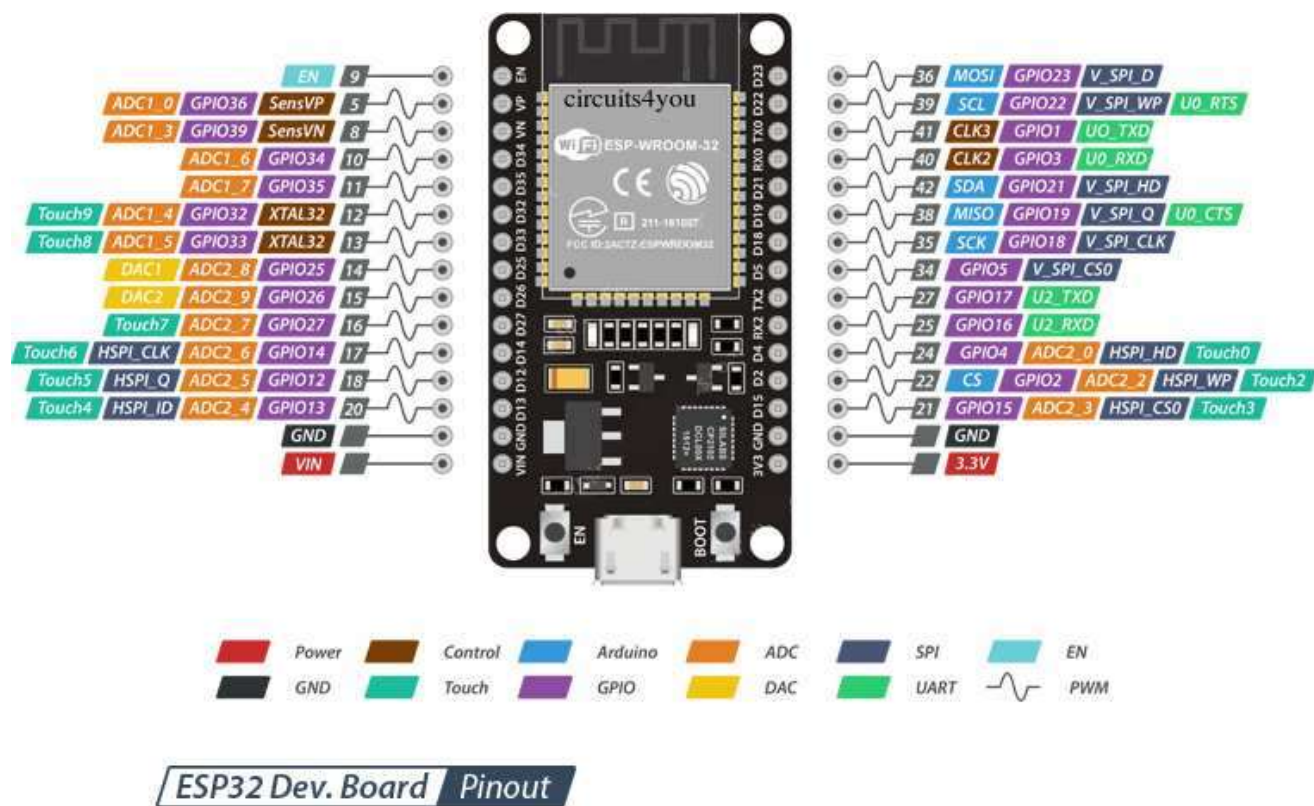


Figure 1 ESP32

### RFID olvasó:

Az RFID (Radio Frequency IDentification) automatikus azonosításhoz és adatközléshez használt technológia, melynek lényege adatok tárolása és továbbítása RFID címkék és eszközök segítségével. Az RFID-címke egy apró tárgy, amely rögzíthető vagy beépíthető az azonosítani kívánt objektumba. Az objektum lehet tárgy, például egy árucikk, vagy alkatrész, illetve élőlény, így akár ember is.

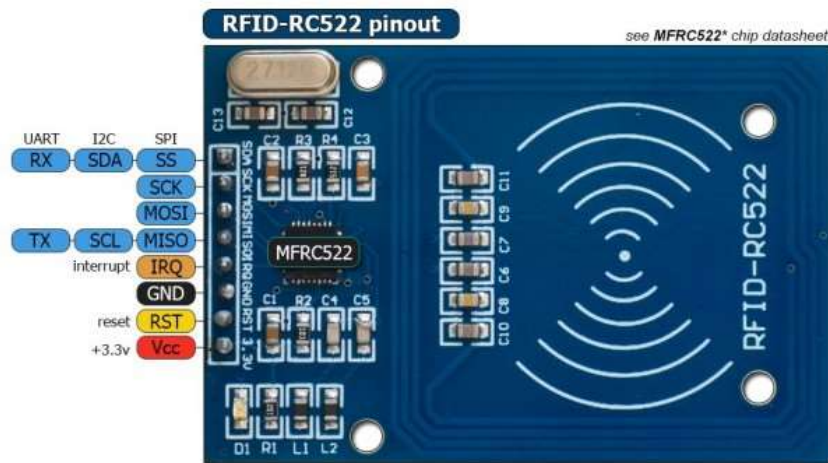


Figure 2 RFID olvasó

### RFID rendszerek működése

1. A szenzor aktiválást indító eseményt érzékel
  2. Az olvasó bekapcsol
  3. Az olvasó leolvassa a bélyegben kódolt adatokat
  4. Az olvasó továbbküldi az adatokat az adatfeldolgozó programot futtató számítógépnek (middleware)
  5. Az adatfeldolgozó számítógép a kiegészített adatokat továbbküldi a vállalatirányítási rendszernek
- Az általam felhasznált kártya olvasó is ezt az elvet követi.

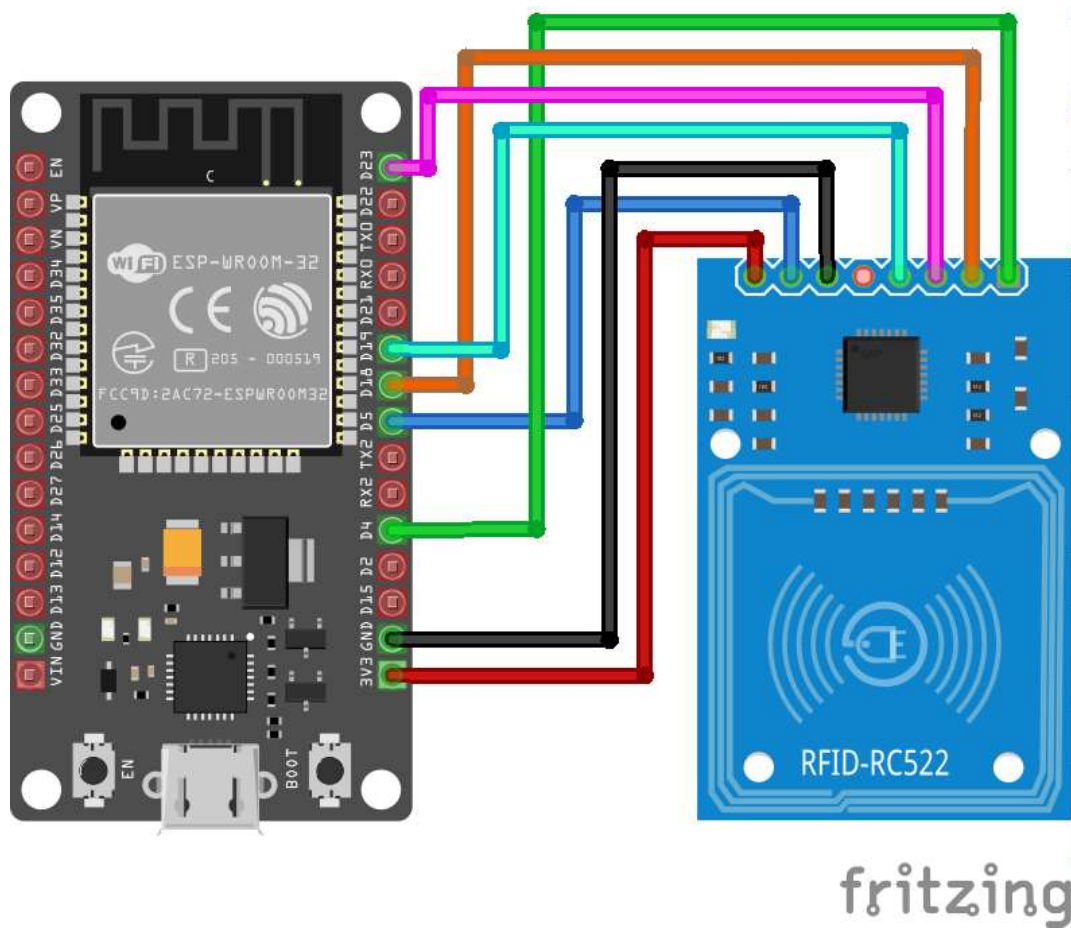


Figure 3 ESP32 és RFID olvasó

A felhasznált arduino program:.

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiMulti.h>
#include <MFRC522.h>
#include <HTTPClient.h>
#include <Arduino_JSON.h>

#define USE_SERIAL Serial
#define SS_PIN    21
#define RST_PIN   22

WiFiMulti wifiMulti;
JSONVar my_elements = JSON.parse("{\"RFID-code\" : \"valami hexa\"}");
MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class
MFRC522::MIFARE_Key key;
byte nuidPICC[4];
String s_rfid="";
```

```

void setup() {
    USE_SERIAL.begin(9600);
    USE_SERIAL.println();
    USE_SERIAL.println();
    USE_SERIAL.println();
    for(uint8_t t = 4; t > 0; t--) {
        USE_SERIAL.printf("[SETUP] WAIT %d...\n", t);
        USE_SERIAL.flush();
        delay(1000);
    }
    wifiMulti.addAP("TP-Link", "asdfghjkl123#");
    SPI.begin(); // Init SPI
    rfid.PCD_Init(); // Init MFRC522
    for (byte i = 0; i < 6; i++) {
        key.keyByte[i] = 0xFF;
    }
}

void loop() {
    if ( ! rfid.PICC_IsNewCardPresent()) return;
    if ( ! rfid.PICC_ReadCardSerial()) return;

    Serial.print(F("PICC type: "));
    MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
    Serial.println(rfid.PICC_GetTypeName(piccType));
    if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
        piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
        piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
        Serial.println(F("Your tag is not of type MIFARE Classic.));
        return;
    }
    for (byte i = 0; i < 4; i++) nuidPICC[i] = rfid.uid.uidByte[i];

    Serial.println(F("The NUID tag is:"));
    Serial.print(F("In hex: "));
    s_rfid="";
    for(byte i=0;i<rfid.uid.size;i++)
    {
        s_rfid+=String(rfid.uid.uidByte[i],HEX);
    }
    Serial.println(s_rfid);
    //printHex(rfid.uid.uidByte, rfid.uid.size);
    Serial.println();
    //setClock();
    // wait for WiFi connection
    if((wifiMulti.run() == WL_CONNECTED)) {

        HTTPClient http;

        USE_SERIAL.print("[HTTP] begin...\n");
        // configure traged server and url
        //http.begin("https://www.howsmyssl.com/a/check", ca); //HTTPS
        http.begin("http://46.40.46.94:81/timers.php?json="+s_rfid); //HTTP

        USE_SERIAL.print("[HTTP] GET...\n");
        // start connection and send HTTP header
        int httpCode = http.GET();

        // httpCode will be negative on error

```

```

        if(httpCode > 0) {
            // HTTP header has been send and Server response header has been
handled
            USE_SERIAL.printf("[HTTP] GET... code: %d\n", httpCode);

            // file found at server
            if(httpCode == HTTP_CODE_OK) {
                String payload = http.getString();
                USE_SERIAL.println(payload);
            }
            } else {
                USE_SERIAL.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
            }

            http.end();
        }

        // Halt PICC
        rfid.PICC_HaltA();

        // Stop encryption on PCD
        rfid.PCD_StopCrypto1();
    }

```

## MySQL adatbázis:.

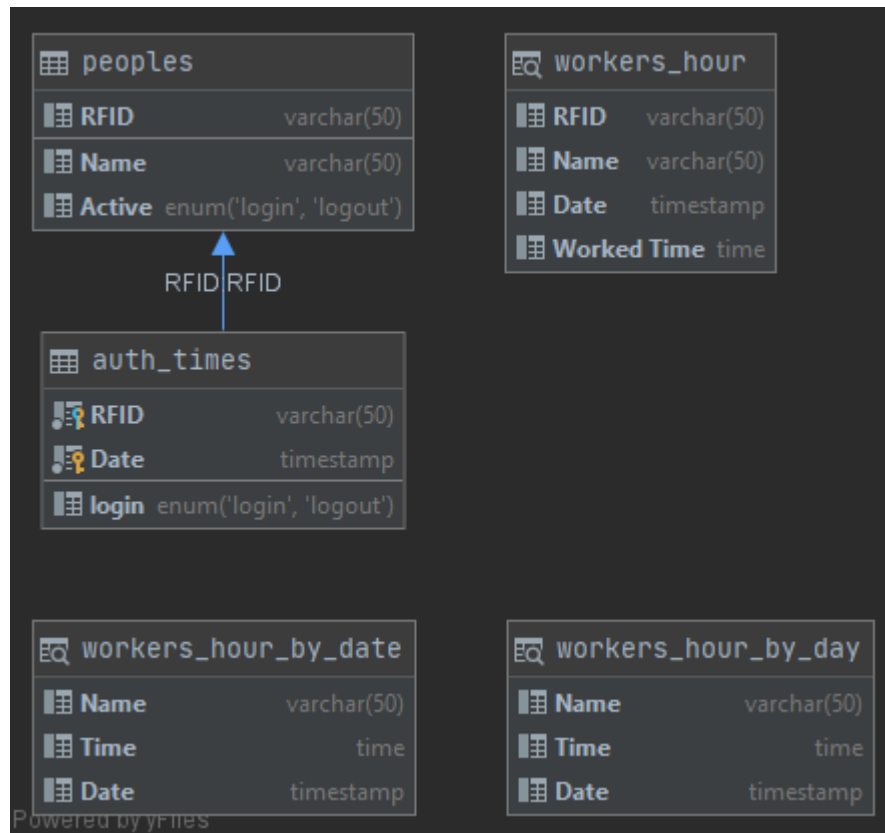


Figure 4 Táblák

A szoftverhoz felhasznált adatbázis kezelő rész.

```
drop procedure if exists auth_times_pre;
delimiter $$
create procedure auth_times_pre(in RFID varchar(50))
begin
    declare c enum('login','logout');
    select Active into c from peoples where peoples.RFID=RFID;
    -- select c;
    if isnull(c) then
        insert into peoples values (RFID,'Temporary Joe',1);
    end if;

    if c='login' then
        -- select 'login';
        insert into auth_times values (RFID,NOW(),'logout');
        update peoples set Active='logout' where peoples.RFID=RFID;
    else
        -- select 'logout';
        insert into auth_times values (RFID,NOW(),'login');
        update peoples set Active='login' where peoples.RFID=RFID;
    end if;
end $$
```



```
DELIMITER ;
```

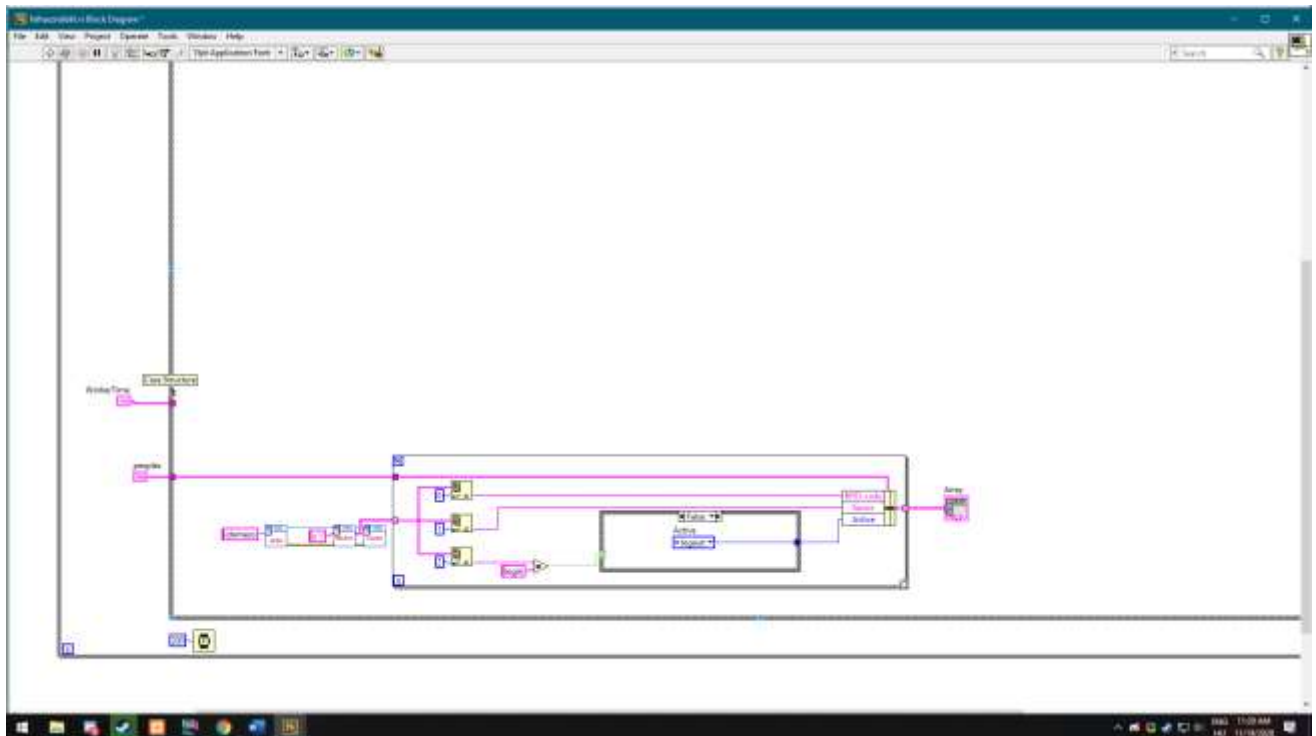
Adatbázisba való adatbevitel.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$conn = new mysqli($servername, $username, $password,"idomero");

//var_dump($_GET);
if(isset($_GET["json"]))
{
    $json=$_GET["json"];
    $res=mysqli_query($conn,"call auth_times_pre('{ $json}')");
    //var_dump($res);
}
```

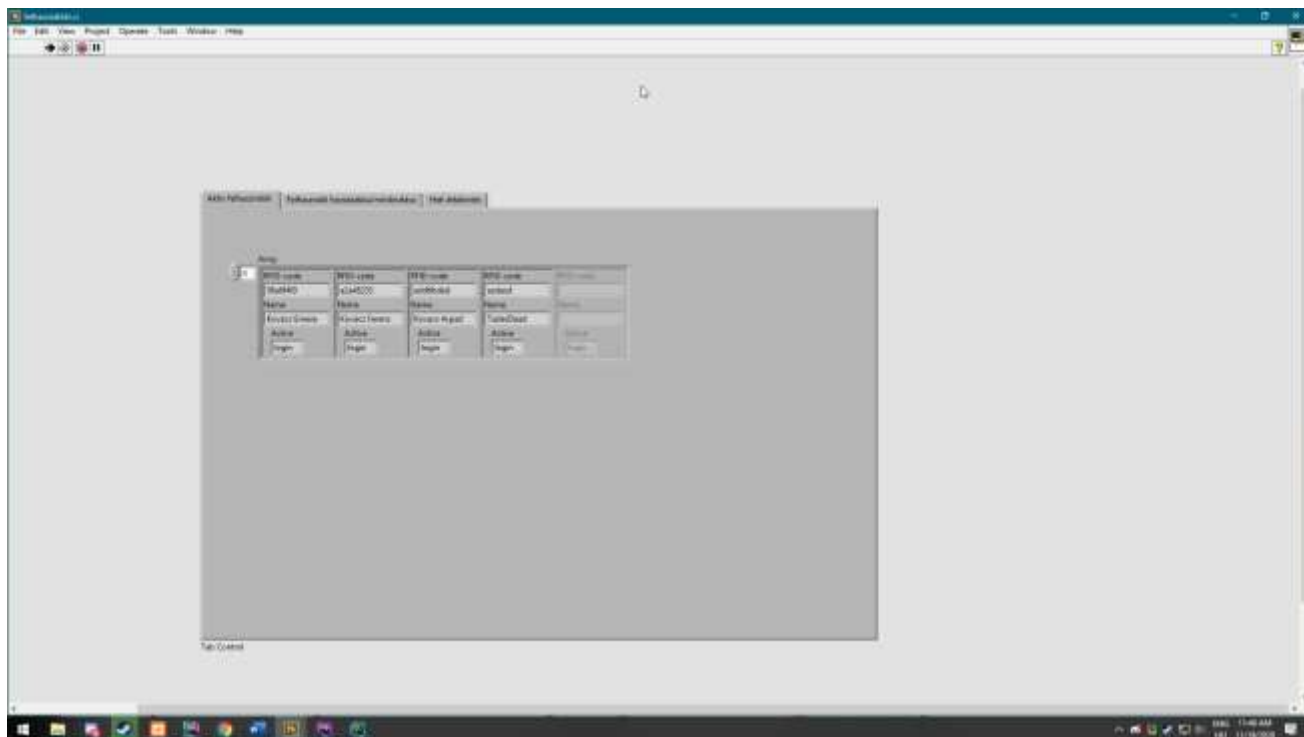
### 3. A LabVIEW szoftver bemutatása

Tartalmazza a szimulációs séma részeit.

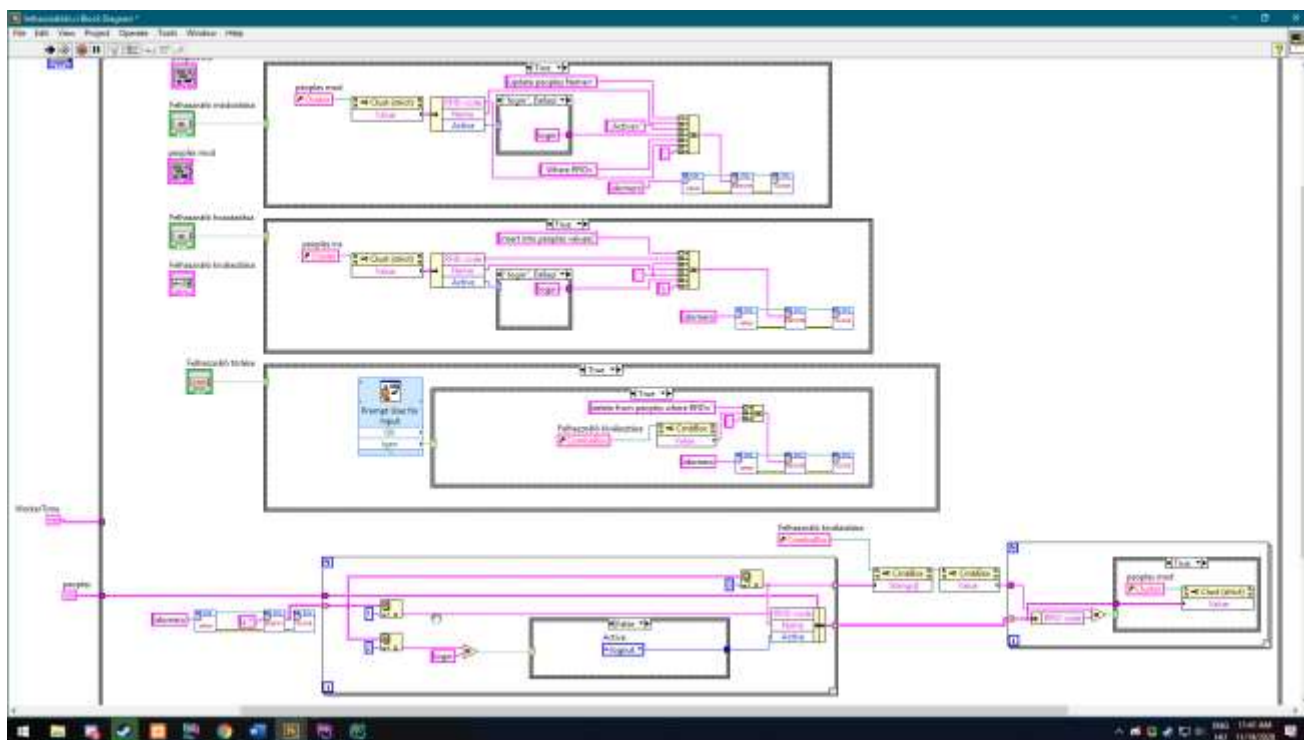


ábra 1 felhasználók áttekintése

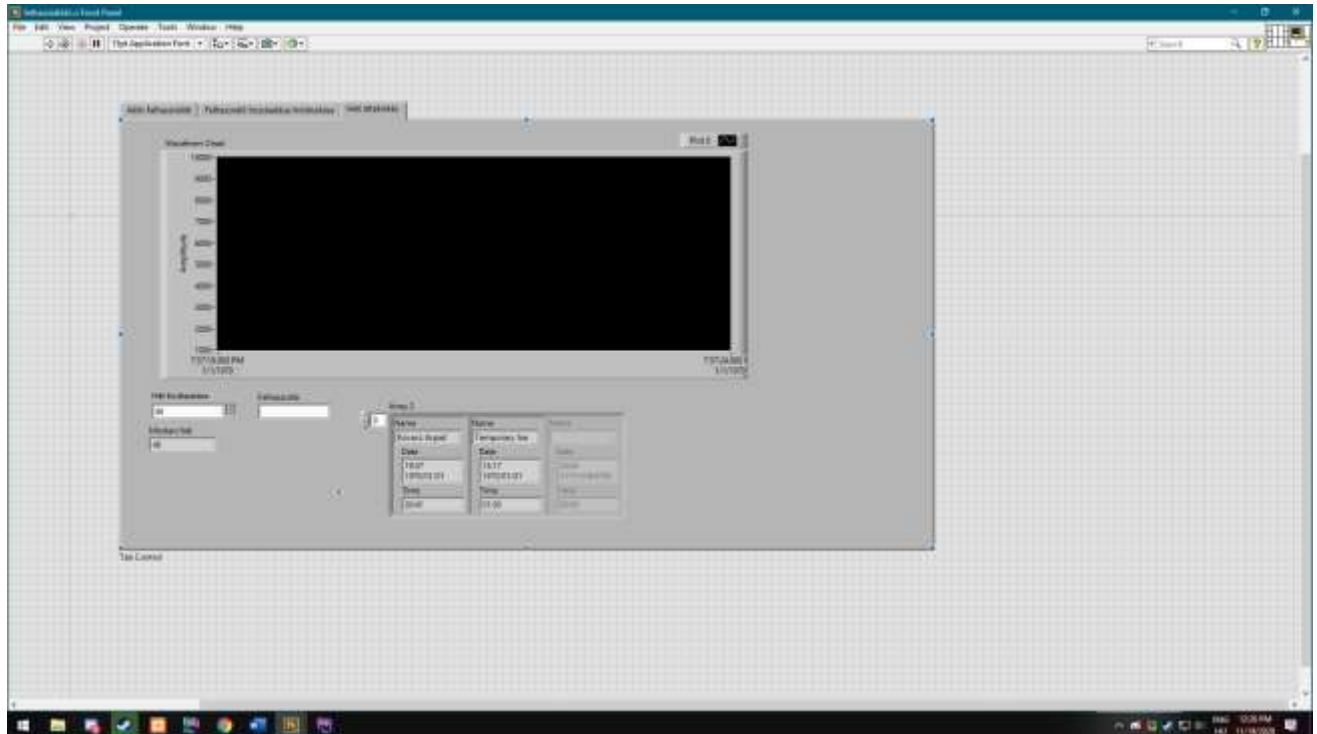
A labview front panelje:.



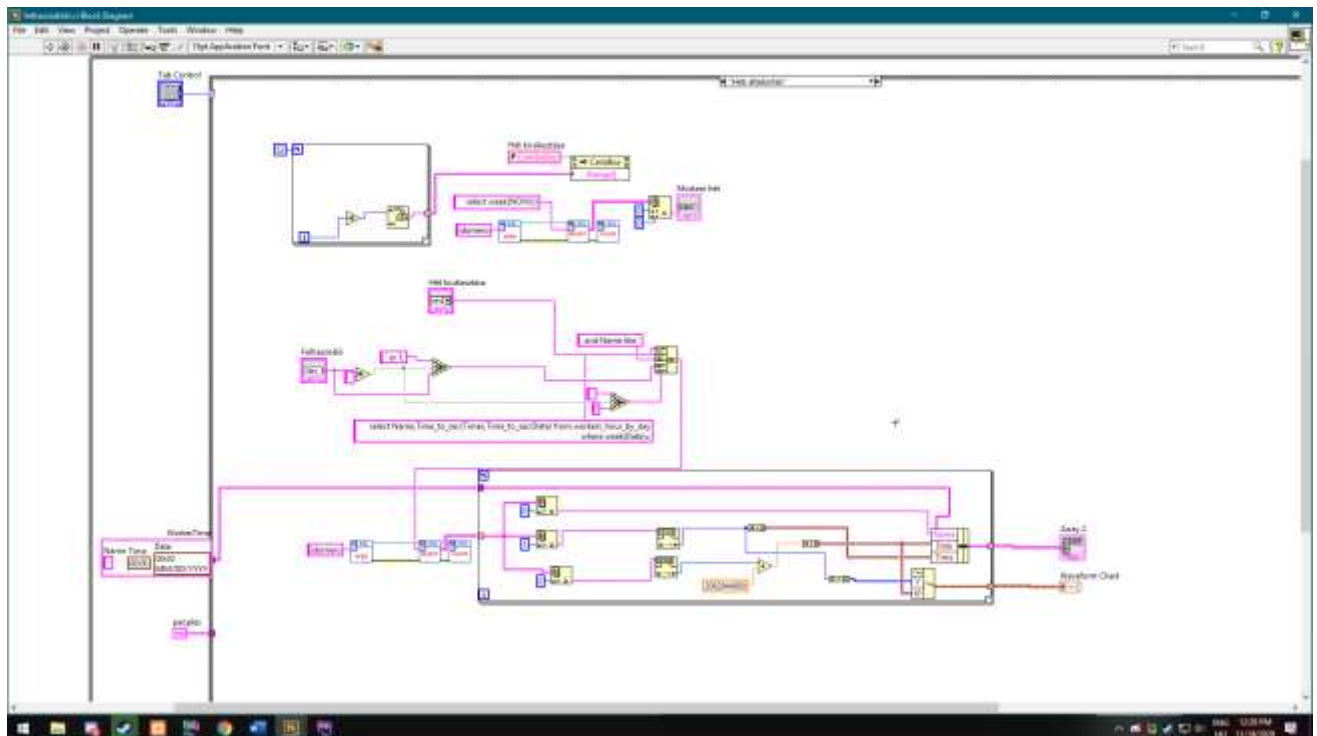
A felhasználók bevitel/módosítása:.



## A felhasználók heti áttekintése



## A felhasználók heti áttekintése sémája:



Kódok:

```
drop view if exists workers_hour;
create view workers_hour as
    select a.RFID,p.Name,a.Date,timediff(b.Date,a.Date) as 'Worked Time' from
auth_times a inner join peoples p on a.RFID = p.RFID
    right join (select RFID,Date,login from auth_times aub order by Date) as b
on a.RFID=b.RFID where a.Date<b.Date and day(b.Date)=day(a.Date) and
a.login='login';
```

Kijelentkezés és bejelentkezés közötti idő mérés:

```
drop view if exists workers_hour_by_day;
create view workers_hour_by_day as
    select Name, SEC_TO_TIME(SUM(TIME_TO_SEC(`Worked Time`)))as Time, Date from
workers_hour group by Name;
```

MySQL Dump

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `idomero`
--

DELIMITER $$
--
-- Procedures
--
CREATE DEFINER=`root`@`localhost` PROCEDURE `auth_times_pre` (IN `RFID`
VARCHAR(50)) begin
    declare c enum('login','logout');
    select Active into c from peoples where peoples.RFID=RFID;
    -- select c;
    if isnull(c) then
        insert into peoples values (RFID,'Temporary Joe',1);
    end if;

    if c='login' then
        -- select 'login';
        insert into auth_times values (RFID,NOW(),'logout');
        update peoples set Active='logout' where peoples.RFID=RFID;
    else
        -- select 'logout';
        insert into auth_times values (RFID,NOW(),'login');
        update peoples set Active='login' where peoples.RFID=RFID;
    end if;

end$$
```

```

DELIMITER ;

-- -----
--
-- Table structure for table `auth_times`
--

CREATE TABLE `auth_times` (
  `RFID` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Date` timestamp NOT NULL DEFAULT current_timestamp(),
  `login` enum('login','logout') COLLATE utf8mb4_unicode_ci DEFAULT 'login'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

--
-- Dumping data for table `auth_times`
--

INSERT INTO `auth_times` (`RFID`, `Date`, `login`) VALUES
('8327bf1a', '2020-11-19 11:17:09', 'login'),
('8327bf1a', '2020-11-19 11:17:26', 'logout'),
('8327bf1a', '2020-11-19 11:18:18', 'login'),
('8327bf1a', '2020-11-19 11:18:20', 'logout'),
('8327bf1a', '2020-11-19 11:24:35', 'login'),
('8327bf1a', '2020-11-19 11:24:39', 'logout'),
('8327bf1a', '2020-11-25 13:25:03', 'login'),
('96a6f4f9', '2020-11-19 11:17:01', 'logout'),
('a2a48233', '2020-11-17 19:25:33', 'login'),
('add6bded', '2020-11-17 17:57:24', 'login'),
('add6bded', '2020-11-17 17:58:48', 'logout'),
('add6bded', '2020-11-17 19:25:06', 'login'),
('add6bded', '2020-11-19 11:16:20', 'logout'),
('add6bded', '2020-11-19 11:16:30', 'login'),
('add6bded', '2020-11-19 11:16:46', 'logout'),
('add6bded', '2020-11-19 11:16:48', 'login'),
('add6bded', '2020-11-19 11:16:54', 'logout'),
('add6bded', '2020-11-19 11:17:56', 'login'),
('add6bded', '2020-11-19 11:17:59', 'logout'),
('add6bded', '2020-11-19 11:18:04', 'login'),
('add6bded', '2020-11-19 11:18:06', 'logout'),
('add6bded', '2020-11-19 11:24:45', 'login'),
('add6bded', '2020-11-19 11:24:55', 'logout');

-- -----
--
-- Table structure for table `peoples`
--

CREATE TABLE `peoples` (
  `RFID` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Name` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Active` enum('login','logout') COLLATE utf8mb4_unicode_ci DEFAULT 'login'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

--
-- Dumping data for table `peoples`
--

```

```

INSERT INTO `peoples` (`RFID`, `Name`, `Active`) VALUES
('8327bf1a', 'Kovács Péter', 'logout'),
('96a6f4f9', 'Kovács Emese', 'logout'),
('a2a48233', 'Kovács Ferenc', 'login'),
('add6bded', 'Kovacs Arpad', 'logout'),
('asdasd', 'TadesDead', 'login');

-- -----

--
-- Stand-in structure for view `workers_hour`
-- (See below for the actual view)
--
CREATE TABLE `workers_hour` (
`RFID` varchar(50)
, `Name` varchar(50)
, `Date` timestamp
, `Worked Time` time
);

-- -----

--
-- Stand-in structure for view `workers_hour_by_date`
-- (See below for the actual view)
--
CREATE TABLE `workers_hour_by_date` (
`Name` varchar(50)
, `Time` time
, `Date` timestamp
);

-- -----

--
-- Stand-in structure for view `workers_hour_by_day`
-- (See below for the actual view)
--
CREATE TABLE `workers_hour_by_day` (
`Name` varchar(50)
, `Time` time
, `Date` timestamp
);

-- -----

--
-- Structure for view `workers_hour`
--
DROP TABLE IF EXISTS `workers_hour`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW
`workers_hour` AS SELECT `a`.`RFID` AS `RFID`, `p`.`Name` AS `Name`,
`a`.`Date` AS `Date`, timediff(`b`.`Date`, `a`.`Date`) AS `Worked Time` FROM
((select `aub`.`RFID` AS `RFID`, `aub`.`Date` AS `Date`, `aub`.`login` AS `login`
from `auth_times` `aub` order by `aub`.`Date`) `b` left join (`auth_times` `a`
join `peoples` `p` on (`a`.`RFID` = `p`.`RFID`)) on (`a`.`RFID` = `b`.`RFID`))
WHERE `a`.`Date` < `b`.`Date` AND dayofmonth(`b`.`Date`) =
dayofmonth(`a`.`Date`) AND `a`.`login` = 'login' ;

```

```

-- -----
--
-- Structure for view `workers_hour_by_date`
--
DROP TABLE IF EXISTS `workers_hour_by_date`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW
`workers_hour_by_date` AS SELECT `tmp`.`Name` AS `Name`, `tmp`.`Time` AS
`Time`, `tmp`.`Date` AS `Date` FROM (select `workers_hour`.`Name` AS
`Name`,sec_to_time(sum(time_to_sec(`workers_hour`.`Worked Time`))) AS
`Time`,`workers_hour`.`Date` AS `Date` from `workers_hour` group by
dayofmonth(`workers_hour`.`Date`)) AS `tmp` GROUP BY `tmp`.`Name` ;

-- -----
--
-- Structure for view `workers_hour_by_day`
--
DROP TABLE IF EXISTS `workers_hour_by_day`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW
`workers_hour_by_day` AS SELECT `workers_hour`.`Name` AS `Name`,
sec_to_time(sum(time_to_sec(`workers_hour`.`Worked Time`))) AS `Time`,
`workers_hour`.`Date` AS `Date` FROM `workers_hour` GROUP BY
`workers_hour`.`Name` ;

--
-- Indexes for dumped tables
--

--
-- Indexes for table `auth_times`
--
ALTER TABLE `auth_times`
  ADD PRIMARY KEY (`RFID`,`Date`);

--
-- Indexes for table `peoples`
--
ALTER TABLE `peoples`
  ADD PRIMARY KEY (`RFID`);

--
-- Constraints for dumped tables
--

--
-- Constraints for table `auth_times`
--
ALTER TABLE `auth_times`
  ADD CONSTRAINT `auth_times_ibfk_1` FOREIGN KEY (`RFID`) REFERENCES `peoples`
(`RFID`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

```

## **A felhasznált rövidítések**

A használt rövidítések jegyzéke és azok jelentése.

RFID-**R**adio **F**requency **ID**entification

MCU-mikrokontroller

## **Irodalom**

[https://www.w3schools.com/php/php\\_mysql\\_intro.asp](https://www.w3schools.com/php/php_mysql_intro.asp)

[https://people.vts.su.ac.rs/~simon/bp2/Baze\\_Podataka2\\_Prirucnik\\_SR\\_2015.pdf](https://people.vts.su.ac.rs/~simon/bp2/Baze_Podataka2_Prirucnik_SR_2015.pdf)

<https://rf-id.lap.hu/>