

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

MASTER ZARIADENIE S DOTYKOVÝM DISPLEJOM
Diplomová práca

2024

Andrej Klein, Bc.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

MASTER ZARIADENIE S DOTYKOVÝM DISPLEJOM
Diplomová práca

Študijný program: Priemyselná elektrotechnika
Študijný odbor: Elektrotechnika
Školiace pracovisko: Katedra elektrotechniky a mechatroniky
Školiteľ: doc. Ing. Milan Lacko, PhD.
Konzultant:

2024 Košice

Andrej Klein, Bc.

Abstrakt v SJ

Táto diplomová práca analyzuje súčasný stav a zapojenie master zariadenia s dotykovým displejom, ktorý je určený pre ovládanie exteriérových žalúzií v inteligentnej domácnosti. Zaoberá sa teoretickou a praktickou časťou tvorby používateľských rozhraní, vizualizácie a programovania mikrokontroléra. Zahŕňa všeobecné informácie a schematické znázornenia potrebné pre návrh a programovanie displeja a master zariadenia. Cieľom práce je dokončiť grafické používateľské prostredie displej a naprogramovať master zariadenie s mikrokontrolérom.

Kľúčové slova v SJ

Inteligentný displej, HMI Nextion, GUI, mikrokontrolér, ATmega324PB, GPIO, USART, TWI/I2C, SPI

Abstrakt v AJ

This thesis analyzes the current status and wiring of a master touchscreen device that is designed to control exterior blinds in a smart home. It deals with the theoretical and practical parts of user interface design, visualization and programming of the microcontroller. It includes general information and schematic representations required for the design and programming of the display and master device. The aim of the thesis is to complete the graphical user interface display and program the master device with a microcontroller.

Kľúčové slova v AJ

Smart external blinds, Intelligent display, HMI Nextion, GUI, microcontroller, ATmega324PB, GPIO, USART, TWI/I2C, SPI

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra teoretickej a priemyselnej elektrotechniky

**ZADANIE
DIPLOMOVEJ PRÁCE**

Študijný odbor: **Elektrotechnika**
Študijný program: **Priemyselná elektrotechnika**

Názov práce:

Master zariadenie s dotykovým displejom
Master device with touchscreen

Študent: **Bc. Andrej Klein**
Školiteľ: **doc. Ing. Milan Lacko, PhD.**
Školiace pracovisko: **Katedra elektrotechniky a mechatroniky**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Oboznámte sa s programovacím prostredím pre dotykový displej Nextion, možnosťou tvorby obrazoviek a komunikáciou s nadradeným mikrokontrolérom
2. Oboznámte sa so zapojením master zariadenia pre komunikáciu s displejom a so spôsobom komunikácie
3. Naprogramujte Nextion displej podľa zadania vedúceho pre aplikáciu ovládania exteriérových žalúzií
4. Naprogramujte Master zariadenie, aby komunikovalo s displejom a následne odosielať príslušné správy pre Slave zariadenia
5. Diplomovú prácu vypracujte podľa pokynov vedúceho (zdrojové kódy sú povinná príloha)

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 19.04.2024
Dátum zadania diplomovej práce: 31.10.2023



n. z. Puhová
.....
prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval/a samostatne s použitím uvedenej odbornej literatúry.

Košice, 30. apríla 2024

.....

vlastnoručný podpis

PodĎakovanie

Moje poĎakovanie patrí hlavne Pánu Bohu za múdrosť a trpezlivosť, ktorú som mal počas celého štúdia na technickej univerzite a tiež pri tvorbe diplomovej práce. Ďalej poĎakovanie patrí všetkým vyučujúcim počas celého inĝinierskeho štúdia a hlavne vedúcemu záverečnej diplomovej práce doc. Ing. Milanovi Lackovi, PhD.

Obsah

Zoznam obrázkov	9
Zoznam tabuliek	11
Zoznam symbolov a skratiek	12
Úvod	13
Formulácia úlohy a cieľ práce.....	14
1. Súčasný stav <i>master</i> zariadenia s dotykovým displejom	15
2. Tvorba používateľských rozhraní	16
2.1. Normy <i>ISO</i> pre tvorbu používateľských rozhraní.....	17
2.2. Používateľské rozhranie dotykového displeja <i>Nextion</i>	17
2.3. Dotykové displeje <i>Nextion</i> všeobecne.....	18
2.3.1. Základná séria	18
2.3.2. Séria <i>Discovery</i>	19
2.3.3. Vylepšená séria	20
2.3.4. Inteligentná séria	21
2.4. Dotykový displej <i>Nextion</i> pre ovládanie exteriérových žalúzií	22
2.4.1. Technické údaje <i>Nextion</i> displeja.....	22
2.4.2. Elektronické charakteristiky <i>Nextion</i> displeja	22
2.4.3. Výkonnosť rozhraní a funkcie pamäte <i>Nextion</i> displeja	22
3. Grafické používateľské rozhranie pre ovládanie exteriérových žalúzií.....	23
3.1. Pracovné prostredie <i>Nextion</i> editora	23
3.2. Vizualizácia.....	26
3.3. Vytvorenie GUI pre ovládanie exteriérových žalúzií.....	27
3.4. Nahrávanie programu do <i>Nextion</i> displeja	33
3.5. Zapojenie <i>Nextion</i> displeja s mikrokontrolérom	33
3.6. Posielanie údajov z <i>Nextion</i> displeja do <i>master</i> zariadenia.....	34
4. <i>Master</i> zariadenie pre ovládanie exteriérových žalúzií.....	37
4.1. Mikrokontrolér <i>ATmega324PU</i>	38

4.1.1.	Zapojenie mikrokontroléra <i>ATmega324PB</i>	40
4.1.2.	Zapojenie tlačidiel a <i>LED</i> diód	41
4.1.3.	Zapojenie senzora reálneho času a dátumu <i>DS1307</i>	42
4.1.4.	Zapojenie senzora teploty a vlhkosti <i>SHT31</i>	43
4.2.	Programovanie <i>master</i> zariadenia	44
4.2.1.	Vstupno-výstupné <i>GPIO</i> porty a <i>CPU</i> hodiny	45
4.2.2.	Programovanie mikrokontroléra pomocou <i>SPI</i> komunikácie	46
4.3.	Štruktúra zdrojového kódu <i>master</i> zariadenia	47
4.3.1.	Program pre <i>master</i> vstupno-výstupné <i>GPIO</i> piny	48
4.3.2.	Program pre <i>master</i> sériovú <i>I2C</i> komunikáciu	49
4.3.3.	Program pre <i>master</i> sériovú <i>USART</i> komunikáciu	52
4.3.4.	Program pre <i>slave</i> sériovú <i>USART</i> komunikáciu	54
4.4.	Overenie zaslaných údajov z <i>master</i> zariadenia do <i>slave</i> zariadení	56
Záver		57
Zoznam použitej literatúry		58
Prílohy		59

Zoznam obrázkov

Obr. 1 HMI dotykový displej [4]	18
Obr. 2 Základná séria <i>Nextion</i> displeja [5]	19
Obr. 3 Séria <i>Discovery Nextion</i> displeja [6]	19
Obr. 4 Vylepšená séria <i>Nextion</i> displeja [7]	20
Obr. 5 Inteligentná séria <i>Nextion</i> displeja [8]	21
Obr. 6 <i>Nextion</i> displej pre ovládanie exteriérových žalúzií	22
Obr. 7 Hlavné rozhranie editora <i>Nextion</i>	23
Obr. 8 Textové, obrázkové, zvukové a video zdroje pre prácu v editore	25
Obr. 9 Hlavná obrazovka <i>GUI</i> exteriérových žalúzií (<i>Page 0</i>)	27
Obr. 10 Povel pre priamy výber exteriérových žalúzií (<i>Page 1</i>)	28
Obr. 11 Nastavenie vybraných exteriérových žalúzií (<i>Page 2</i>)	29
Obr. 12 Hláška pre užívateľa pri úspešnej komunikácii.	29
Obr. 13 Výber budíkov, režim umývania okien a zrušenie manuálneho režimu (<i>Page 3</i>)	30
Obr. 14 Výber žalúzie pre budík1, 2 a 3 (<i>Page 5, 7 a 9</i>)	30
Obr. 15 Nastavenie budíka 1, 2 a 3 (<i>Page 6, 8 a 10</i>)	31
Obr. 16 Nastavenie dátumu, dňa, času a podsvietenia displeja (<i>Page 4</i>)	32
Obr. 17 Nahrávanie TFT kódu cez <i>microSD</i> priamo do displeja	33
Obr. 18 Zapojenie HMI displeja s MCU cez <i>UART</i>	34
Obr. 19 Zapojený HMI displej <i>Nextion</i> s master zariadením	34
Obr. 20 Odosielanie údajov z displeja cez <i>UART</i>	34
Obr. 21 Master zariadenie s čipom <i>ATmega324PB</i>	37
Obr. 22 Mikrokontrolér <i>ATmega324PB</i> z externým 32kHz kryštálom	38
Obr. 23 Úplne zapojenie <i>DPS master</i> zariadenia	39
Obr. 24 Elektrická schéma obvodu mikrokontroléra s obvodom externého kryštálu	40
Obr. 25 Elektrická schéma resetovacieho (vľavo) a napájacieho obvodu (vpravo)	40
Obr. 26 Elektrická schéma obvodov pre tlačidlá a <i>LED</i> diódy	41
Obr. 27 Elektrická schéma obvodu <i>RTC</i> modulu <i>DS1307</i>	42
Obr. 28 Funkčná bloková schéma <i>DS1307</i> [14]	42
Obr. 29 Elektrická schéma snímača relatívnej teploty a vlhkosti <i>SHT31</i>	43
Obr. 30 Funkčná bloková schéma <i>SHT31</i> [15]	43
Obr. 31 Vstupno-výstupné piny mikrokontroléra <i>ATmega324PB</i>	44
Obr. 32 Vnútna štruktúra vstupno-výstupných pinov mikrokontroléra	45
Obr. 33 Zapojenie <i>SPI</i> prevodníka <i>USB-ASP</i> s MCU	46

Obr. 34 Štruktúra zdrojového kódu pre <i>master</i> zariadenie	47
Obr. 35 Zdrojový kód funkcií pre odoslanie čísla po sériovej <i>USART</i> komunikácií.....	53
Obr. 36 Zapojenie <i>USART</i> prevodníka <i>USB/TTL</i> s <i>MCU</i>	56
Obr. 37 Softvér <i>PUTTY</i>	56
Obr. 38 Výstup odoslaných dát <i>master</i> zariadenia <i>slave</i> zariadeniam cez <i>TTL</i> (<i>PUTTY</i> terminál) ...	56

Zoznam tabuliek

Tab. 1 Komunikačné kódy a povely displeja s <i>master</i> zariadením	35
Tab. 2 Zoznam súčiastok <i>DPS master</i> zariadenia	39
Tab. 3 <i>RTC</i> adresy s umiestnením bitov pre hodnoty dátumu a času v <i>DS1307</i>	50
Tab. 4 Kódy príkazov pre <i>slave</i> zariadenia.	55

Zoznam symbolov a skratiek

ARM	Advanced RISC Machine
DPS	doska plošných spojov
EDL	Event Driven Language
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPIO	General Purpose Input / Output
GUI	Graphical User Interface
HMI	Human Machine Interface
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
ISO	International Organization for Standardization
LCD	Liquid Crystal Display
MCU	Microcontroller unit
MMI	Man Machine Interface
PWM	Pulse Width Modulator
RAM	Random Access Memory
RISC	Reduced instruction set computing
RTC	Real Time Clock
SPI	Serial Peripheral Interface
TFT	Thin Film Transistor
TTL	Transistor-transistor logic
TWI	Two-Wire Interface
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter

Úvod

V súčasnej dobe je automatizácia inteligentných domácností stále viac prítomná. A to najmä v oblasti *smart* technológií a riadenia domácností pomocou užívateľských rozhraní alebo aplikácií. Do tejto kategórie inteligentných domácností spadá aj riadenie vonkajších žalúzií. Efektívne a pohodlné ovládanie týchto zariadení môže prispieť k zvýšeniu komfortu užívateľov a optimalizácii energetickej efektívnosti domu. V tejto práci sa zameriavame na vývoj a implementáciu systému riadenia vonkajších žalúzií pomocou *Nextion* displeja a *master* zariadenia s mikrokontrolérom *ATmega324PB*. V práci je uvedený súčasný stav *master* zariadenia s dotykovým displejom, ktorý popisuje v akom stave, bola táto práca prevzatá.

Práca je venovaná teoretickým informáciám používateľských rozhraní, ako sú *MMI*, *ISO* normy, *Nextion* displeje všeobecne, vizualizácia, tvorba *GUI* a samotný mikrokontrolér *ATmega324PB*. Okrem toho je v práci uvedený postup overovania údajov zo sériovej komunikácií pomocou *TTL*.

Praktická časť diplomovej práce obsahuje vytvorenie a nahrávanie *GUI* do *Nextion* displeja s odosielaním údajov do *master* zariadenia. Okrem toho obsahuje hlavnú časť, programovanie *master* zariadenia podľa štruktúry zdrojového kódu, kde sú vysvetlené zdrojové kódy pre snímače používajúce *I2C* a *USART* komunikáciu.

Sústredíme sa na analýzu, návrh a implementáciu tejto technológie. Cieľom môže byť vytvorenie užívateľsky prívetivej a spoľahlivej platformy na riadenie vonkajších žalúzií, ktorá je prispôbená pre domácnosti a malé komerčné objekty.

Formulácia úlohy a cieľ práce

Hlavným cieľom tejto práce je naprogramovať dotykový displej pomocou IDE rozhrania *Nextion* editora a upraviť jednotlivé obrazovky, ktoré boli už predpripravené. Taktiež nastaviť a doprogramovať funkcie a premenné s ktorými pracujú jednotlivé obrazovky. Ďalej je potrebné doprogramovať funkčné tlačidlá v jednotlivých obrazovkách tak, aby po stlačení vysielali nastavené parametre, ako povely do mikrokontroléra *ATmega32* po sériovej linke, následne tieto povely mikrokontrolér spracuje a odošle ich do *slave* zariadení po druhej sériovej linke.

Prvá kapitola je venovaná súčasnemu stavu *master* zariadenia s dotykovým displejom.

Druhá kapitola analyzuje grafické používateľské rozhrania a dotykové *Nextion* displeje všeobecne. Poukazuje na technické parametre, výkonnosť a série displejov.

Tretia kapitola tvorí návrh grafického používateľského rozhrania pre projekt ovládania exteriérových žalúzií. V kapitole je popísané pracovné prostredie *Nextion* editora. V kapitole sa tiež rieši nahrávanie programu do displeja a komunikácia s *master* zariadením.

Štvrtá kapitola je zameraná pre naprogramovanie a komunikáciu *master* zariadenia. V kapitole sa rieši mikrokontrolér *ATmega324PB*, *GPIO*, *RTC* modul pre reálny dátum a čas, snímač teploty a vlhkosti a komunikácia so *slave* zariadeniami. Poukazuje na zapojenie a program.

1. Súčasný stav *master* zariadenia s dotykovým displejom

Nextion dotykový displej a vývojová doska *DPS master* zariadenia (mikrokontrolér *ATmega324PB*) sú hardvérovo k dispozícii. Je potrebné spojazdniť komunikáciu potrebnú pre programovanie *master* zariadenia, pričom je nevyhnutné dokúpiť *SPI-ASP* programátor, ktorým budeme nahrávať naprogramované zdrojové kódy do *master* zariadenia, cez *SPI* zbernicu. K úplnej výbave pre vypracovanie zadania nám postačí už len *notebook* s nainštalovaným softvérom pre prácu s displejom a *master* zariadením.

Čo sa týka užívateľského prostredia *Nextion* displeja a grafiky obrazoviek, tak je táto práca čiastočne rozpracovaná a z väčšej časti je návrh obrazoviek a parametrov hotový, preto je najprv potrebné oboznámiť sa s užívateľským prostredím a dokončiť niektoré obrazovky pomocou konzultácií s vedúcim projektu.

Master zariadenie je hardvérovo pripravený, ale je potrebné naprogramovať za pomoci registrov a rozbehať všetky potrebné periférie, a to sú *USART* zbernica, *I2C/TWI* zbernica, *AD/GPIO* periféria. Po úspešnej komunikácii daných zberníc je potrebné naprogramovať samotné moduly reálneho dátumu/času a snímača teploty/vlhkosti (*DS1307* a *SHT31*), ktoré pomocou týchto zberníc komunikujú.

Po úspešnej komunikácii periférií v *master* zariadení je potrebné naprogramovať samotnú logiku reálneho času a dátumu s funkciou nastavovania parametrov. Tento modul bude ovládaný aj pomocou samotného displeja *Nextion*. Podobné požiadavky sa vyžadujú aj od snímača teploty a vlhkosti s tým rozdielom, že snímač teploty a vlhkosti slúži len na čítanie dát.

Po naprogramovaní vyššie spomenutých periférií, modulov a úspešnej komunikácii s displejom, posledná vec, ktorú je potrebné rozbehať a naprogramovať je samotné odosielanie údajov do *slave* zariadení pomocou druhej *USART* zbernice v *master* zariadení. Tieto údaje, ktoré sú odosielané *master* zariadením, je potrebné overiť napríklad pomocou *USB-TTL* prevodníka.

2. Tvorba používateľských rozhraní

Táto kapitola sa zaoberá definíciou, charakteristikou používateľského rozhrania, normami a technickými parametrami *Nextion* displejov. V kapitole sú predstavené všetky série displejov, ktoré *Nextion* ponúka.

Činnosť návrhu rozhrania si okrem programátorských zručností vyžaduje aj určité vedomosti týkajúce sa psychologického a ergonomického hľadiska návrhu systémov *MMI* (*Man Machine Interface*). Disciplína *MMI* študuje komunikáciu medzi strojmi a ľuďmi. [1]

Návrh užívateľského rozhrania je činnosť zameraná na vývoj rozhrania aplikácií používaných v počítačoch, strojoch, mobilných zariadeniach, internetových prehliadačoch alebo displejoch ako je *Nextion*. Návrh sa zameriava na efektivitu a ergonómiu užívateľskej interakcie s ohľadom na zabezpečenie kvality, spoľahlivosti a bezpečnosti. [1]

Proces tvorby užívateľského rozhrania musí vytvoriť rovnováhu medzi technickou funkcionalitou a vizuálnymi elementmi, tak aby vytvoril systém, ktorý je nie len funkčný, ale aj jednoducho a logicky použiteľný pre užívateľov aplikácie. Zámerom používateľského rozhrania je uľahčiť prenos informácií medzi používateľom a systémom, ktorý má byť riadený. [1]

Systém *MMI* tvorí súhrn programových a technických možností, ktoré pomáhajú maximálnemu zjednodušeniu obsluhy riadeného systému a tiež zabezpečujú: [1]

- zobrazenie stavu procesu pomocou grafických prostriedkov,
- operátorské riadenie procesu,
- archiváciu priebehu výroby a činnosti obsluhy,
- sledovanie a spracovanie alarmových situácií,
- sledovanie a riadenie kvality produkcie,
- sledovanie, plánovanie a údržba zariadení.

2.1. Normy ISO pre tvorbu používateľských rozhraní

Táto podkapitola sa zaoberá štruktúrou verzie slovenskej technickej normy STN EN ISO 9241, ktorá má postavenie národnej normy, ktorá bola schválená Slovenským ústavom technickej normalizácie. Stručne opisuje častí, ktoré sú zamerané na tvorbu používateľských rozhraní. [1]

Norma **STN EN ISO 9241** je prínosom pre návrhárov používateľského rozhrania. Princípy uplatnené v praxi bude využívať používateľ, ktorého potreby sú určené ergonomickými požiadavkami pri tvorbe noriem. Aplikácia normy by mala viesť k vytvoreniu používateľského rozhrania, ktoré vedie k zvýšeniu produktivity práce používateľa. [1]

Norma **STN EN ISO 9241** pozostáva zo 17 častí, z ktorých časti 10, 13, 14 a 17 sú zamerané na tvorbu používateľských rozhraní: [1]

- ISO 9241-10:1996 - Základné zásady vytvárania dialógu.
- ISO 9241-13:1998 - Príručka používateľa.
- ISO 9241-14:1997 - Vedenie dialógu pomocou menu.
- ISO 9241-17:1998 - Vedenie dialógu pomocou obrazových formulárov.

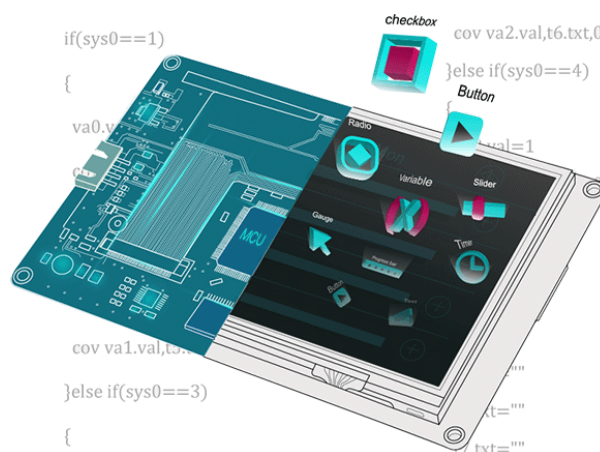
V úvode je popísaný predmet normy s odkazmi na súvisiace normy a vysvetľuje definície s použitými pojmami. Všetky časti obsahujú aplikovateľnosť a vhodnosť použitia konkrétnej časti normy v daných podmienkach. Jadro obsahuje zoznam odporúčaní, ktoré sú špecifické pre všetky časti normy v daných podmienkach. Niekedy sú odporúčania popísané praktickými príkladmi. [1]

2.2. Používateľské rozhranie dotykového displeja *Nextion*

Inteligentný displej spracováva informácie na základe vnímania okolitého prostredia alebo užívateľských vstupov. Niektoré inteligentné displeje môžu mať aj integrovanú umelú inteligencia alebo strojové učenie, čo im umožňuje prispôbovať sa potrebám používateľa alebo prostredia. [2]

HMI (Human-Machine Interface) je používateľské rozhranie, ktoré spája človeka so strojom, systémom alebo zariadením. *HMI* zahŕňa rôzne technológie, ako sú obrazovky dotykových displejov, ovládacie prvky a softvérové rozhrania, ktoré umožňujú používateľom komunikovať s automatizovanými systémami. [2]

Grafické používateľské rozhrania (*GUI*) sa často využívajú v rámci *HMI* na vizualizačné možnosti, ako je monitorovanie a sledovanie, alebo zložitejšie operácie riadenia systémov. *GUI* dotykového displeja umožňuje interakciu medzi používateľom a mikropočítačom prostredníctvom vizuálnych ovládacích prvkov, ako sú ikony, okná, tlačidlá, menu, textové polia, formuláre a rôzne iné ovládacie prvky. Do budúcnosti sa skúmajú spôsoby implementácie rozšírenej reality (*AR*) a virtuálnej reality (*VR*) na vizualizáciu výrobných funkcií. [3]



Obr. 1 *HMI* dotykový displej [4]

2.3. Dotykové displeje *Nextion* všeobecne

Nextion je riešenie rozhrania *HMI*, ktoré kombinuje vstavaný procesor a pamäťový dotykový *TFT* displej so softvérom *Nextion Editor* na vývoj projektu *GUI HMI*. *Nextion* zaisťuje kontrolu a vizualizáciu rozhrania medzi človekom a strojom, systémom alebo zariadením. [4]

Tieto displeje sú populárne práve pre ich jednoduchú integráciu do rôznych projektov. Majú užívateľský prívetivé rozhranie, ktoré umožňuje jednoduché vytváranie *GUI* bez potreby detailného programovania. Sú vybavené *TTL* (*transistor-transistor logic*) sériovým portom, ktorý sa pripája k periférnemu *MCU* (*Microcontroller unit*), cez *UART* (*Universal asynchronous receiver-transmitter*) zbernicu, pre komunikáciu s mikrokontrolérmi alebo inými riadiacimi zariadeniami. Napájajú sa vstupným jednosmerným 5V napätím z externého zdroja. [4]

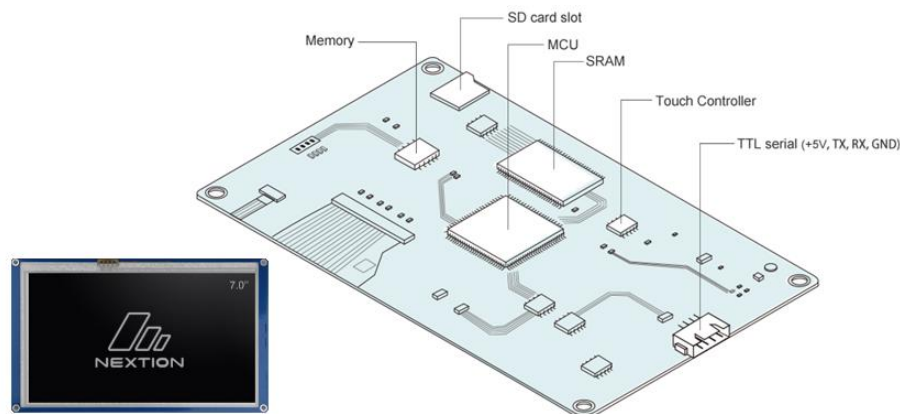
Nextion displeje sa používajú prevažne v oblasti *IoT* (*Internet of things*). Tieto dotykové *TFT LCD* obrazovky sú dostupné v 4 sériách a rôznych veľkostiach.

2.3.1. Základná séria

Nextion displeje základnej série ponúkajú univerzálne veľkosti obrazovky od 2,4" do 7" s rozlíšením od 320x240 pixelov až do 800x480 pixelov, čo poskytuje dostatočnú kvalitu obrazu. Displeje sú vybavené 48 MHz mikroprocesorom, ktorý umožňuje samostatné fungovanie bez

potreby externého *MCU*. *Nextion* používa odporovú technológiu pre dotykovú funkciu, čo znamená, že reaguje na fyzický tlak na povrchu obrazovky. Odporová *TFT* dotyková obrazovka má až 65536 RGB farieb. [5]

Nextion displeje obsahujú 3584 bajtov pamäte *RAM* a 4 až 16 MB internej *Flash* pamäte pre ukladanie grafických prvkov, textov, čísiel a iných dát na zobrazenie *GUI* pre *HMI* projekt. Majú zabudovaný *SD* slot pre nahrávanie programu a uloženie dát. [5]

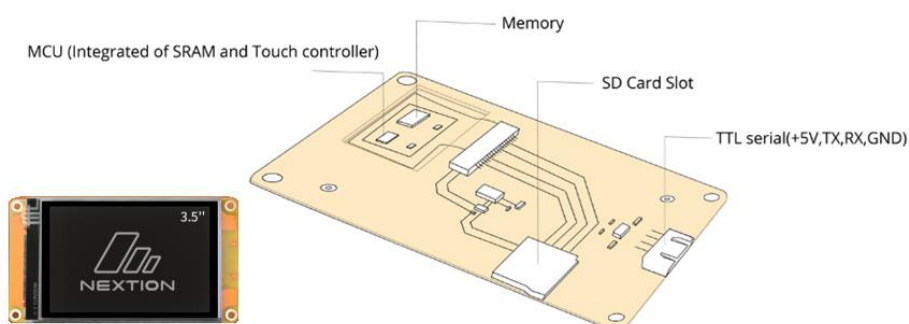


Obr. 2 Základná séria *Nextion* displeja [5]

2.3.2. Séria *Discovery*

Nextion displeje série *Discovery* ponúkajú tri možnosti veľkosti obrazovky 2,4", 2,8" a 3,5" s rozlíšením 320x240 pixelov a 480x320 pixelov. Displeje sú vybavené až 64 MHz mikroprocesorom, čo je výkonnejšie ako pri základnej sérii. [6]

Nextion displeje obsahujú 3584 bajtov pamäte *RAM* a 4 až 16 MB internej *Flash* pamäte ako pri základnej sérii. Majú zabudovaný *SD* slot pre nahrávanie programu a uloženie dát. [6]



Obr. 3 Séria *Discovery* *Nextion* displeja [6]

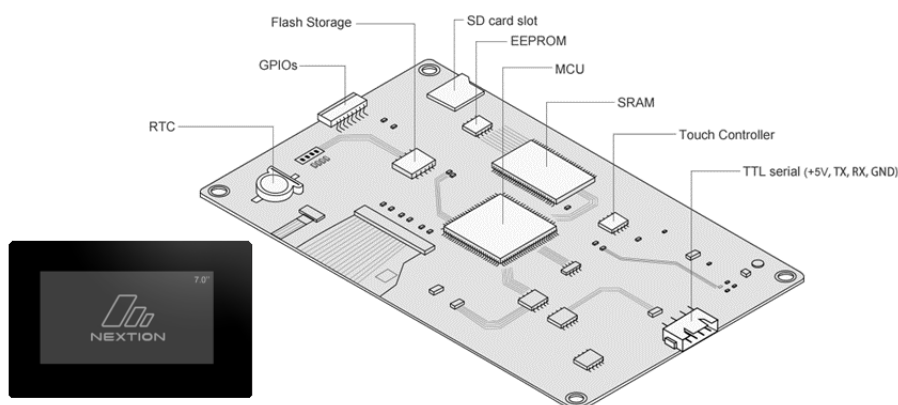
Odporová *TFT* dotyková obrazovka má rovnako 65536 RGB farieb ako pri základnej sérii. Prevádzkový prúd v režime hlbokého spánku je len 0,25 mA, vďaka čomu sú displeje radu *Discovery* vhodnejšie pre projekty napájané z batérie. [6]

2.3.3. Vylepšená séria

Nextion displeje vylepšenej série ponúkajú univerzálne veľkosti obrazovky od 2,4" do 7" s rozlíšením od 320x240 pixelov až do 800x480 pixelov, čo poskytuje dostatočnú kvalitu obrazu. Na rozdiel od základnej série sú displeje vybavené od 48 MHz do 108 MHz mikroprocesorom, čo je výkonnejšie ako pri základnej sérii. [7]

Tieto *Nextion* displeje obsahujú od 3584 bajtov do 8192 bajtov pamäte *RAM* a 16 až 32 MB internej *Flash* pamäte pre ukladanie grafických prvkov, textov, čísiel a iných dát na zobrazenie *GUI* pre *HMI* projekt. Tak isto ako v základnej sérii majú zabudovaný *SD* slot pre nahrávanie programu a uloženie dát. [7]

Okrem toho majú integrovanú 1024 bajtovú pamäť *EEPROM* na ukladanie dát používateľov, 8 digitálnych *GPIO* (*General-Purpose Input/Output*) s podporou štyroch *PWM* (*Pulse width modulation*) a *RTC* (*Real Time Clock*) na udržiavanie presného času. [7]

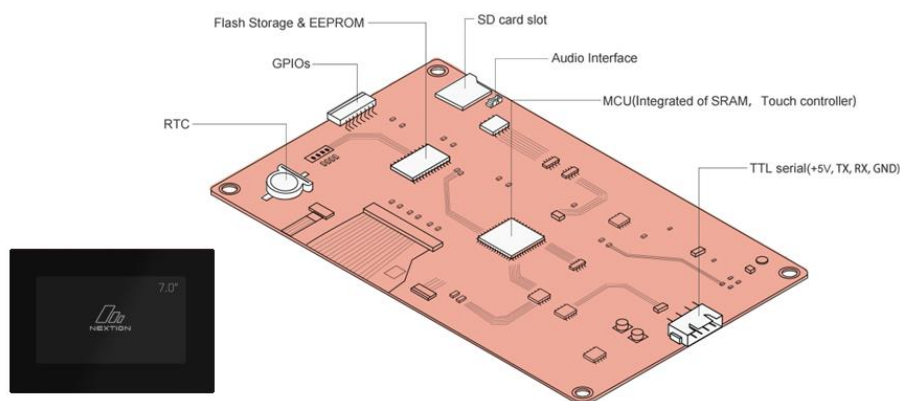


Obr. 4 Vylepšená séria *Nextion* displeja [7]

Odporová *TFT* dotyková obrazovka má rovnako 65536 RGB farieb, ako pri základnej sérii. Vylepšená séria na rozdiel od základnej už ponúka aj kapacitnú 7" dotykovú obrazovku s krytom. [7]

2.3.4. Inteligentná séria

Nextion displeje inteligentnej série ponúkajú 4 veľkosti obrazovky 4,3", 5", 7" a 10,1" s rozlíšením 480x272 pixelov, 800x480 pixelov a 1024x600 pixelov čo poskytuje dostatočnú kvalitu obrazu. Na rozdiel od predošlých sérií sú displeje vybavené 200 MHz mikroprocesorom s 512 kB pamäťou *RAM* a 128 MB internou *Flash* pamäťou pre ukladanie grafických prvkov, textov, čísiel a iných dát na zobrazenie *GUI* pre *HMI* projekt, tiež s možnosťami rôznych exkluzívnych funkcií a vlastností pre sériu *intelligent*. Tak isto ako v základnej sérii majú zabudovaný *SD* slot pre nahrávanie programu a uloženie dát . [8]



Obr. 5 Inteligentná séria *Nextion* displeja [8]

Podobne ako u vylepšenej sérií majú integrovanú 1024 bajtovú pamäť *EEPROM* na ukladanie dát používateľov, 8 digitálnych *GPIO* s podporou štyroch *PWM* a *RTC* na udržiavanie presného času. [8]

Odporová *TFT* dotyková obrazovka má rovnako 65536 RGB farieb. Na rozdiel od predošlých sérií všetky displeje inteligentnej série ponúkajú aj kapacitnú dotykovú obrazovku. [8]

Novinkou je pridané audio rozhranie. Funkcie prehrávania zvuku, videa a animácie obohacujú interakciu *HMI* projektu používateľa. Inteligentná séria podporuje pokročilé softvérové funkcie, ako je transparentný komponent, efekt načítania stránky, komponent *Move and Drag* atď. [8]

2.4. Dotykový displej *Nextion* pre ovládanie exteriérových žalúzií

Pre náš projekt ovládania exteriérových žalúzií je zvolený 5" dotykový displej *Nextion* s rozlíšením 800x480 pixelov základnej série typu NX8048T050.



Obr. 6 *Nextion* displej pre ovládanie exteriérových žalúzií

2.4.1. Technické údaje *Nextion* displeja

Displej je vybavený 48 MHz mikroprocesorom, ktorý umožňuje samostatné fungovanie bez potreby externého *MCU*. Displej obsahuje 3584 bajtov pamäte *RAM* a 16 MB internej *Flash* pamäte pre ukladanie grafických prvkov, textov, čísiel a iných dát na zobrazenie *GUI* pre náš *HMI* projekt. [5][9]

Má zabudovaný *SD* slot pre nahrávanie programu a uloženie dát. Dotyková obrazovka má 65536 RGB farieb. Podsvietenie je typu *LED*, kde priemerná životnosť podsvietenia vydrží viac ako 30000 hodín. Jas je 230 nit. [5][9]

2.4.2. Elektronické charakteristiky *Nextion* displeja

Prevádzkové napätie je v rozmedzí od 4,75 V do 7 V. Prevádzkový prúd, kde napätie obrazovky je 5 V a jas 100 % dosahuje 410 mA. *Nextion* displej pracuje aj v režime spánku, kde elektrický prúd dosahuje 15 mA. Odporúčané elektrické napájanie obrazovky výrobcom je jednosmerných 5 V, 1 A. [9]

2.4.3. Výkonnosť rozhraní a funkcie pamäte *Nextion* displeja

Displej nemá *USB* rozhranie, ale zabudovaný slot na *SD* kartu (formát *FAT32*), ktorý podporuje maximálne 32G *Micro SD* kartu. Používa sa výhradne na aktualizáciu firmvéru. [9]

Režim sériového portu typu *TTL* pozostáva so 4 pinov (*VCC*, *RX*, *TX* a *GND*).Štandardná prenosová rýchlosť sériového portu môže byť 2400 bps, 9600 bps alebo 115200 bps. [9]

Nextion displej využíva na ukladanie textov, písma a obrázkov 16 MB *Flash* pamäť. Na ukladanie premenných využíva 3584 bytovú pamäť *RAM*. [9]

3. Grafické používateľské rozhranie pre ovládanie exteriérových žalúzií

Najväčšou výhodou dotykových displejov *Nextion* je grafický editor, v ktorom sa dá ľahko a bez problémov nadefinovať grafické a dotykové prostredie a vykonať simulácie. [10]

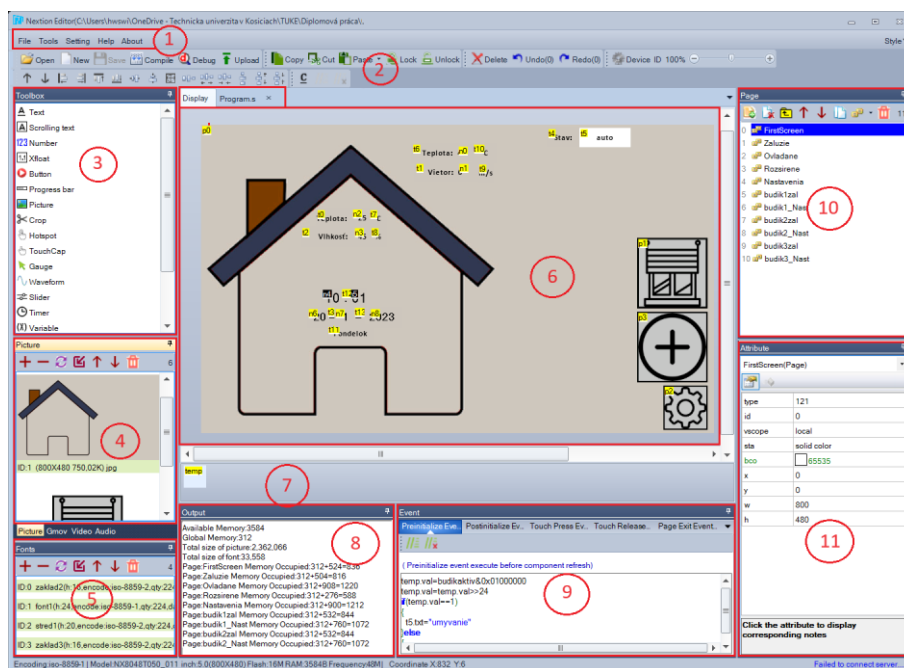
Nextion editor je bezplatný vývojový softvér, ktorý sa používa na rýchle vytváranie používateľských rozhraní GUI „človek-stroj“ pre dotykové displeje *Nextion HMI*. [10]

Obsahuje nástroje na grafické navrhovanie rozhraní, pridávanie interaktívnych prvkov a programovanie funkcionality pomocou jednoduchého programovacieho jazyka *EDL (Event-Driven Language)*. Rýchle vytvorenie *HMI GUI* pre *Nextion* nevyžaduje extrémne zručnosti no očakávajú sa aspoň základné zručnosti v programovaní. [11]

Záťaž späť s grafikou je priamo prenesená do *ARM* procesora displeja.

3.1. Pracovné prostredie Nextion editora

Ako prvá vec pri tvorbe *GUI* pre zvolený displej, je vytvorenie nového projektu, kde sa nastavil presný typ displeja s jeho konkrétnou uhlopriečkou a fyzickým natočením. Na nasledujúcom obrázku je zobrazené hlavné rozhranie editora *Nextion* s otvoreným projektom pre exteriérové žalúzie.



Obr. 7 Hlavné rozhranie editora *Nextion*

Pre objasnenie v nasledujúcom texte je vysvetlené hlavné rozhranie editora:

1 – hlavné menu, 2 – panely nástrojov, 3 – panel s nástrojmi, 4 – panel zdrojov obrázkov, 5 – panel zdrojov písma, 6 – dizajnové plátno aktuálnej obrazovky (vizuálne komponenty) a hlavný program, 7 – nevizuálne komponenty, 8 – výstup, 9 – kód udalosti používateľa, 10 – panel všetkých obrazoviek (*Page*), 11 – panel atribútov.

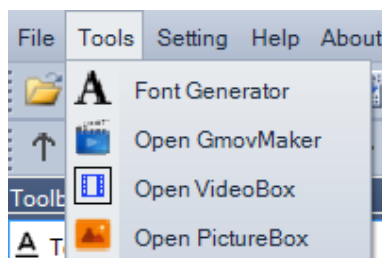
Pomocou hlavného menu (1) je možné vytvoriť nové projekty, otvoriť alebo importovať existujúce projekty. Tiež uložiť vytvorené *HMI* projekty a generovať výstupný *TFT* súbor, ktorý sa nahráva priamo do *Nextion* displeja pomocou *micro SD* karty. V ponuke nástrojov je možnosť prístupu k externým nástrojom, ako sú generátor fontov, *GmovMaker*, *VideoBox* a *PictureBox*. Ponuka nastavení obsahuje dve položky a to konfiguráciu a obnovenie rozloženia. Ponuka pomoci obsahuje taktiež dve položky a to inštrukčný set a sprievodcu editorom s odkazom na web. Pomocou inštrukčného setu bola vytvorená logika programovania a všetky funkcie pracujúce s displejom. V inštrukčnom sete sa nachádza kompletná inštrukčná sada pre používanie všetkých príkazov editora *Nextion*. [10]

Z panela nástrojov (2) sa využíva hlavne kompilátor a funkcia *debug*, taktiež sa využíva možnosť priameho nahrávania naprogramovaného *TFT* výstupu do displeja cez UART komunikáciu. Pomocou kompilátora sa kontrolujú chyby zdrojových kódov v celom projekte. Všetky varovania alebo chybové hlásenia sa umiestnia na výstupný panel (8). Kompilácia je proces budovania a tvorby výstupného kódu, preto programátori musia venovať svoju pozornosť všetkým varovným a chybovým hláseniam. *Nextion* editor obsahuje taktiež vstavaný simulátor, ku ktorému sa pristupuje práve pomocou funkcie *debug*. Tento emulátor je určený na pomoc programátorovi pri ladení používateľského rozhrania *GUI*. [10]

Panel s nástrojmi (3) poskytuje užívateľovi prístup k rôznym komponentom, nástrojom a funkciám, ktoré sú potrebné pre návrh a programovanie *GUI*. Obsahujú grafické prvky ako sú tlačidlá, textové polia, číselné polia, *slider*, obrázky, animácie, časovače, premenné a ďalšie. [10]

Všetky obrázky používané v projekte je potrebné importovať cez panel zdrojov obrázkov (4). Ešte pred začiatkom vytvorenia projektu musí programátor vytvoriť fonty písma a textu, ktoré budú v užívateľskom rozhraní využívané pomocou panela zdrojov písma (5). [10]

Niektoré fonty sú v samotnom vývojovom prostredí zadefinované, ale programátor má možnosť tvorby vlastných fontov pomocou generátora fontov v hlavnom menu (1) v častí nástroje (obr. 8). *Nextion* editor má tiež možnosť importovania aj *Gmov* zdrojov (animácie), zvukových a video zdrojov. [10]



Obr. 8 Textové, obrázkové, zvukové a video zdroje pre prácu v editore

Dizajnové plátno (6) je hlavný dizajnový priestor pre vizuálne/dotykové komponenty pre aktuálnu stránku. V tejto častí sa nachádza aj hlavný program „*Program.s*“, ktorý si programátor vie prepínať s dizajnovým plátnom „*Display*“. Pod plátnom sa nachádza časť s nevizuálnymi komponentami stránky (7), sú to premenné alebo časovače, bez ktorých sa programátor nezaobíde. Na stránku je povolený limit 250 komponentov (vizuálnych a nevizuálnych). [10]

Výstup (8) obsahuje podrobnosti o procese zostavovania výstupného kódu *TFT* pri možnostiach kompilácie, *debug-u* alebo nahrávania do displeja. Kód udalosti používateľa (9) môže obsahovať akúkoľvek platnú inštrukciu alebo časť naprogramovaného kódu, ktorý je vždy lokálny pre aktuálnu stránku. Takmer každý komponent má udalosti „*Touch Press*“ – čo sa udeje pri stlačení dotyku na displej a „*Touch Release*“ – čo sa udeje pri pustení dotyku z displeja. [10]

Každý projekt *HMI* musí mať aspoň jednu grafickú stránku „*Page*“. Stránky je možné vytvárať a importovať do projektu *HMI* prostredníctvom panela stránok (10). Stránky sú odlišované číselne. Panel atribútov (11) obsahuje zoznam komponentov zahrnutých na stránke aktuálneho dizajnu. Kliknutím na komponent sa zobrazia dostupné atribúty komponentu. Atribúty čiernej farby sú len na čítanie počas behu programu a atribúty zelenej farby sú na čítanie aj zmenu počas behu programu. [10]

3.2. Vizualizácia

Vizuálna prezentácia je najdôležitejšou časťou *GUI* rozhrania medzi procesmi, automatickými riadiacimi systémami a používateľmi. Vizualizácia je zameraná pre užívateľov s cieľom vyvolať u nich vizuálne dobrý pocit (vizuálny vnem). [1]

Pomocou vizualizácie používame teoretické, technické, programové a komunikačné prostriedky pre zviditeľnenie objektov informačného a riadiaceho systému s cieľom podpory rozhodovania. Vizualizácia sa netýka len grafického zobrazovania objektov, ale jedná sa o všetky činnosti získavania, definície a spracovávaní objektov, ktoré sú grafickou stránkou *GUI* rozhraním medzi priemyselnými procesmi a používateľmi. [1]

Ciele procesov používateľských vizualizácií sú: [1]

- monitorovanie, zber a záznam údajov, stavov alebo procesov,
- sledovanie činnosti systému a overovanie funkcií,
- ovládanie, riadiaca a monitorovacia činnosť systému,
- diagnostika procesu a predvídanie chybových stavov.

Vizualizácia priemyselných procesov je rozhranie medzi technologickými časťami a informačnými úrovňami riadiacich systémov. [1]

Základné funkcie vizualizačného systému sú: [1]

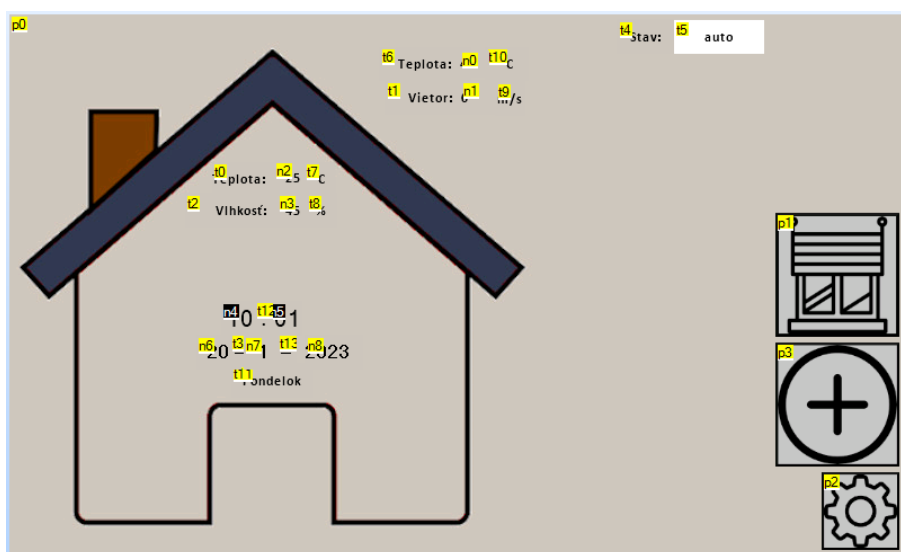
- prenášanie povelov používateľov pre technologické procesy,
- prenášanie a zobrazenie používateľom údaje o reálnom stave,
- hodnoty stavových veličín, parametrov alebo okamžité štruktúry,
- zbieranie a uchovávanie informácií o stavoch technologického procesu (archivovanie),
- zobrazovanie alebo uchovávanie informácií o poruche systému (alarm),
- zobrazovanie vývoja daných veličín v konkrétnom časovom období (trendy).

3.3. Vytvorenie GUI pre ovládanie exteriérových žalúzií

Po vytvorení nového projektu pre zvolený displej a zadefinovaní všetkých nastavení displeja (typ displeja, uhlopriečka a natočenie), boli postupne vytvorené nasledujúce obrazovky, ktoré sa zobrazujú v *Nextion* displeji. Na každej obrazovke sú vytvorené rôzne funkcie, ktoré používajú vlastné nástroje, ako sú tlačidlá, obrázky, *slide* alebo obyčajné texty.

V hlavnej obrazovke *Page 0* (obr. 9) exteriérových žalúzií je znázornené intuitívne rozhranie, ktoré zobrazuje aktuálny dátum pomocou funkcií *n6*, *n7*, *n8* a *t11* displeja a čas pomocou funkcií *n4* a *n5* displeja. Dátum a čas sú premenné, ktoré je možné meniť užívateľom. Okrem toho rozhranie *Page 0* zobrazuje údaje o vnútornej teplote pomocou funkcie *n2*, vlhkosti pomocou funkcie *n3*, vonkajšej teplote pomocou funkcie *n0* a rýchlosti vetra pomocou funkcie *n1* v prehľadnom formáte.

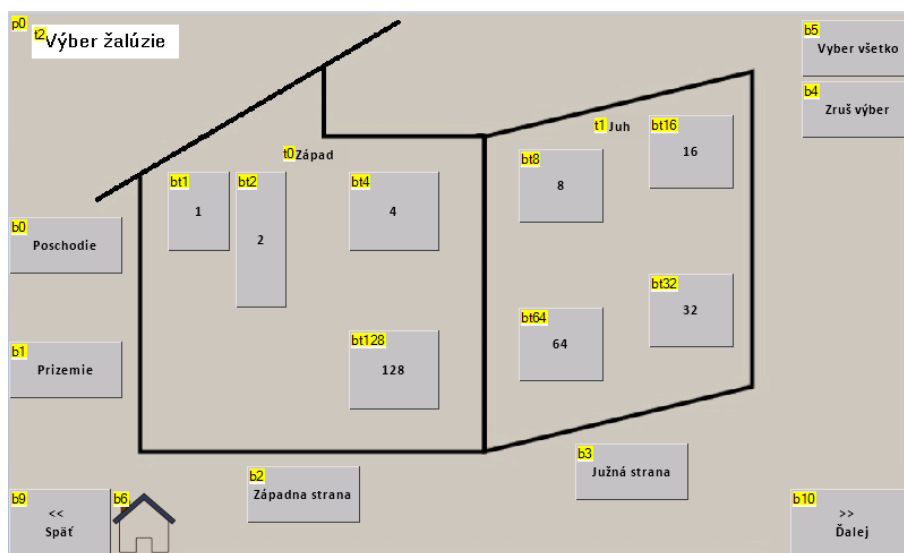
Hlavná obrazovka *Page 0* zobrazuje stav zariadenia pomocou funkcie *t5*, či je v manuálnom, automatickom režime alebo v režime umývania okien. Obrazovka taktiež ponúka prítomnosť hlavného menu pomocou funkcií *p1*, *p3* a *p2* na pravej strane, ktoré umožňujú prístup k ďalším obrazovkám a funkciám.



Obr. 9 Hlavná obrazovka *GUI* exteriérových žalúzií (*Page 0*)

Ak užívateľ stlačí v hlavnom menu prvú možnosť, to je tlačidlo *p1*, tak vstúpi do ďalšej obrazovky *Page 1*.

Táto obrazovka je vytvorená pre priame zadanie povelov *slave* zariadeniam s možnosťou výberu žalúzií pre konkrétne okná. Na tejto obrazovke si vie užívateľ jednoducho a rýchlo zvoliť žalúzie, pre ktoré budú platiť nastavenia, ktoré sú v ďalšej obrazovke po zakliknutí tlačidla „Ďalej“. Užívateľ má možnosť voľby celého poschodia, prízemí, západnej strany a južnej strany jedným kliknutím. Tiež môže jedným kliknutím zvoliť všetky žalúzie alebo zrušiť zvolený výber žalúzií. Pre rýchli návrat na hlavnú obrazovku slúži tlačidlo „b6“ s ikonou domčeka.



Obr. 10 Poveľ pre priamy výber exteriérových žalúzií (Page 1)

Tlačidlo b10 „Ďalej“ sa zobrazí, ak je zvolené aspoň jedno okno (tlačidlá bt1 až bt128). Program je navrhnutý tak, že do jednej premennej sa zapíšu všetky okná pomocou bitových operácií. Program je navrhnutý nasledovne:

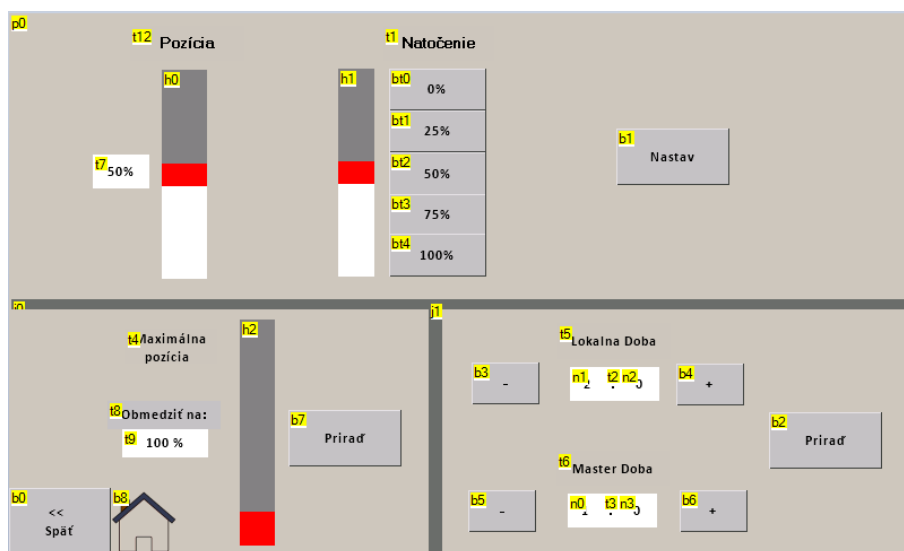
```

zaluzielD.val = bt128.val
zaluzielD.val = zaluzielD.val<<1
zaluzielD.val += bt64.val
zaluzielD.val = zaluzielD.val<<1
zaluzielD.val += bt32.val
zaluzielD.val = zaluzielD.val<<1
zaluzielD.val += bt16.val
zaluzielD.val = zaluzielD.val<<1
zaluzielD.val += bt8.val
zaluzielD.val = zaluzielD.val<<1
zaluzielD.val += bt4.val
zaluzielD.val = zaluzielD.val<<1
zaluzielD.val += bt2.val
zaluzielD.val = zaluzielD.val<<1
zaluzielD.val += bt1.val

```

Pomocou tejto logiky sú v dvoj bajtovej premennej „zaluzielD“ zahrnuté všetky okná/žalúzie. Táto premenná bude odosielaná cez UART zbernicu, ako jedno hexadecimálne číslo.

Po zakliknutí tlačidla „Ďalej“ sa dostane užívateľ do spomínaného nastavenia vybraných exteriérových žalúzií v obrazovke *Page 2* (obr. 11). Užívateľ si môže nastaviť priamym povelom pozíciu (výšku) a natočenie žalúzií pomocou tlačidla „Nastav“. Okrem toho môže určiť maximálnu pozíciu žalúzií, *lokálnu* dobu a *master* dobu pomocou tlačidla „Prirad“. Tieto dáta sa ihneď po kliknutí odošlú do *master* zariadenia, ktorý tieto dáta spracuje a odošle ďalej *slave* zariadeniam pre konkrétne žalúzie. Pre rýchli návrat na hlavnú obrazovku slúži tlačidlo „b8“ s ikonou domčeka.



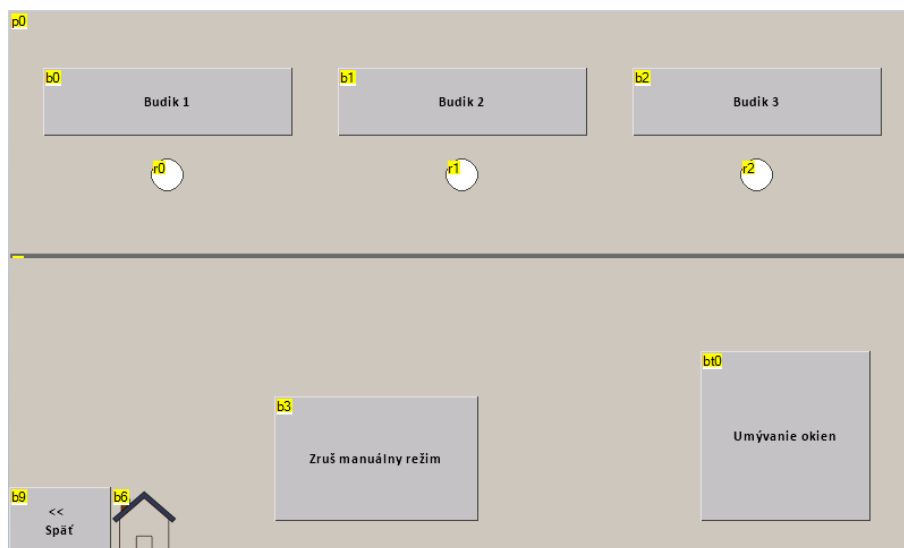
Obr. 11 Nastavenie vybraných exteriérových žalúzií (*Page 2*)

Všetky povely z displeja sa odosielaajú cez *UART* zbernicu, aby nedošlo k nedorozumeniu medzi užívateľom a *master* zariadením pri strate údajov, ktoré neprišli z displeja do *master* zariadenia. Tento možný problém je ošetrený chybovou hláškou, kde pri úspešnej komunikácii sa zobrazí užívateľovi nasledujúca správa (obr. 12).



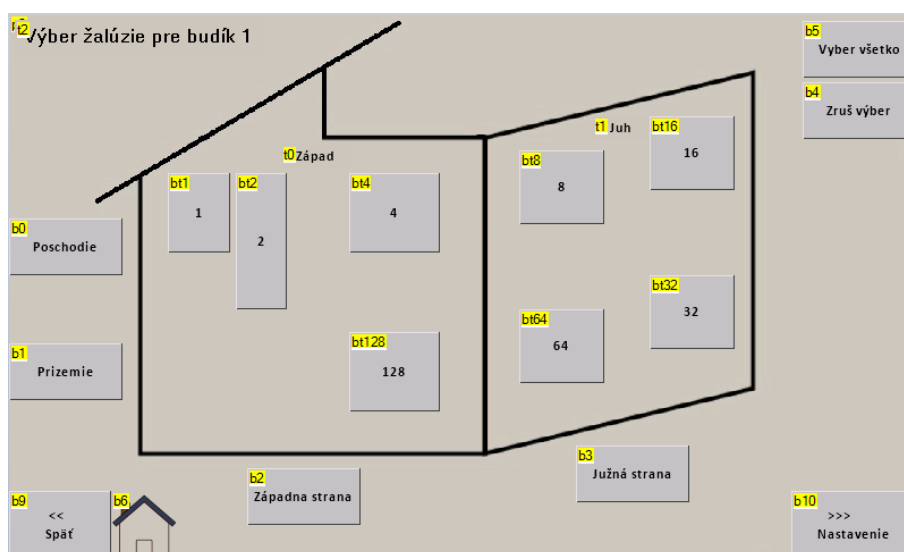
Obr. 12 Hláška pre užívateľa pri úspešnej komunikácii.

Ak užívateľ stlačí v hlavnom menu (obr. 9) druhú možnosť a to tlačidlo *p3*, tak vstúpi do ďalšej obrazovky *Page 3* (obr. 13), kde má možnosť voľby budíkov, zapnutia režimu pre umývanie okien alebo zrušenia manuálneho režimu. Tieto povely sa ihneď po kliknutí odošlú do *master* zariadenia, ktorý ich spracuje a odošle ďalej *slave* zariadeniam, ako povel.



Obr. 13 Výber budíkov, režim umývania okien a zrušenie manuálneho režimu (*Page 3*)

Pri voľbe budíkov stlačením tlačidiel pomocou funkcií *b0*, *b1* alebo *b2* sa užívateľ dostane na ďalšiu obrazovku (obr. 14), kde vyberie pre ktoré žalúzie má byť daný budík nastavený. Rozhranie je podobné, ako u priameho zadania povelov v obrazovke *Page 1* (obr. 10). Užívateľ si môže aktivovať až 3 budíky nezávislé od seba. Pre rýchli návrat na hlavnú obrazovku slúži tlačidlo „b6“ s ikonou domčeka.



Obr. 14 Výber žalúzie pre budík1, 2 a 3 (*Page 5, 7 a 9*)

Po zvolení vybraných žalúzií a stlačení tlačidla „*Nastavenie*“ pomocou funkcie *b10* sa užívateľ dostane k nastaveniu konkrétneho budíka, kde si nastaví na akú pozíciu sa žalúzie po zobudení nastaví, taktiež natočenie žalúzií po zobudení a samozrejme čas budenia kedy sa má daná akcia vykonať. Okrem času budenia má užívateľ možnosť nastavenia dňa pre ktorý bude budík platiť. Sú predpripravené 3 varianty budíkov s ktorými vie užívateľ pracovať.



Obr. 15 Nastavenie budíka 1, 2 a 3 (Page 6, 8 a 10)

Nastavenie budíkov je iba lokálna funkcia, ktorá sa po stlačení tlačidla „*Nastav budík*“ pomocou funkcie *b10* uloží v pamäti premenných displeja. Pre rýchli návrat na hlavnú obrazovku slúži tlačidlo „*b6*“ s ikonou domčeka.

Pomocou funkcií prepínacích tlačidiel „*radio*“ *r0*, *r1* a *r2* na obrazovke *Page 3* (obr. 13) sa ihneď aktivuje budík a odošle informácia do *master* zariadenia o stave a plynutí daného budíka, ktorý tuto udalosť vyhodnocuje a následne komunikuje so *slave* zariadeniami.

Ak užívateľ stlačí v hlavnom menu (obr. 9) tretiu možnosť a to tlačidlo *p2*, tak vstúpi do ďalšej obrazovky *Page 4* (obr. 16), ktorá je vytvorená pre nastavenia s ktorými bude pracovať užívateľ. Tu má možnosť nastaviť dátum, deň a čas displeja. Tieto nastavenia sa po stlačení tlačidla „NASTAV“ pomocou funkcie *b13* ihneď odošlú do *master* zariadenia, ktorý aktualizuje dátum, deň a čas. Keďže opäť sa odosiela správa cez *UART* zbernicu, tak komunikácia je ošetrená overovacou hláškou (obr. 12).

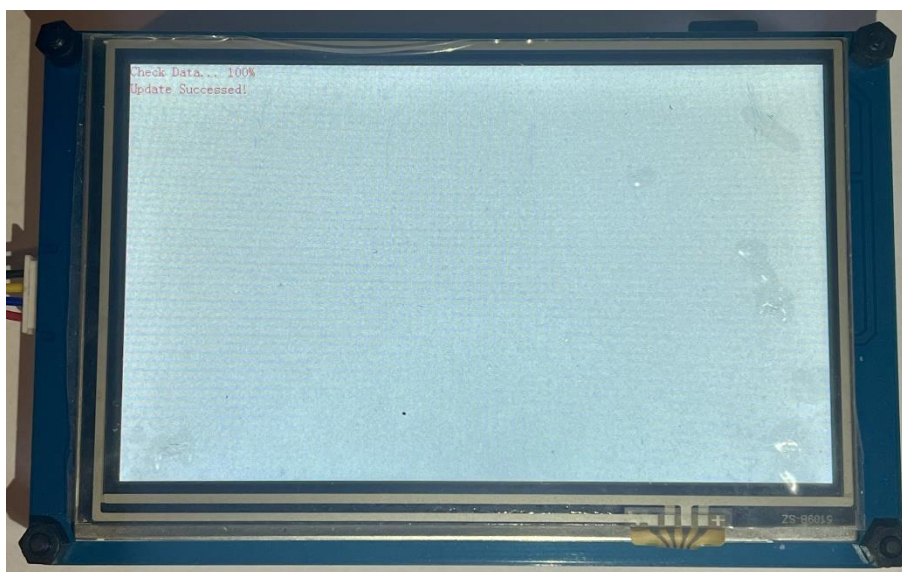
Obr. 16 Nastavenie dátumu, dňa, času a podsvietenia displeja (*Page 4*)

Okrem nastavenia dátumu, dňa a času si užívateľ môže nastaviť na tejto obrazovke aj podsvietenie displeja tlačidlami pomocou funkcií *b8* a *b10* (obr. 16). Pre rýchli návrat na hlavnú obrazovku slúži tlačidlo „b16“ s ikonou domčka.

3.4. Nahrávanie programu do *Nextion* displeja

Nahrávanie dokončeného naprogramovaného projektu *Nextion HMI* je možné vykonať buď pomocou *microSD* karty alebo, cez *UART* rozhranie. Ak použijeme na nahrávanie *UART* zbernicu, tak musí byť displej prepojený pomocou *TTL-USB* prevodníka a nastavený *baud-rate*. Základnou požiadavkou pri nahrávaní pomocou *microSD* karty je, aby bola vo formáte *FAT32*. [11]

Po úspešnom skompilovaní *HMI* kódu, bol vyexportovaný výstupný súbor *TFT* na kartu *microSD*. Po vložení karty do slotu *SD* v *Nextion* displeji a po zapnutí displeja na napájacie napätie sa automaticky rozoznala požiadavka pre zápis programu a nahral sa súbor *TFT* do internej pamäte *Flash* v *Nextion* displeji. [11]



Obr. 17 Nahrávanie TFT kódu cez *microSD* priamo do displeja

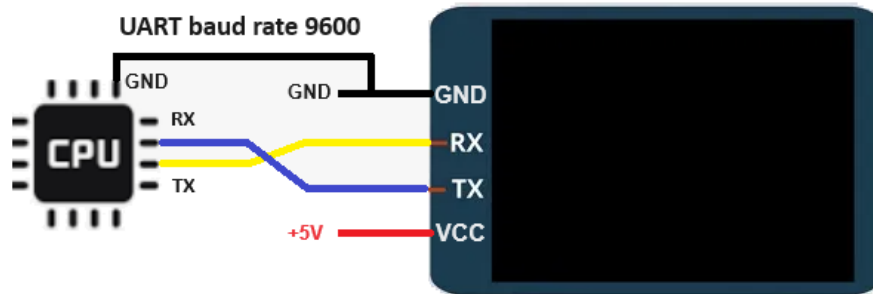
Po úspešnom nahrať *TFT* programu do displeja, bol odpojený displej od napájania a *SD* karta vytiahnutá zo slotu *SD*. Program sa takto automaticky nahral a je uložený v samotnej pamäti displeja. Po opätovnom pripojení napájacieho napätia do displeja sa zobrazí vytvorené užívateľské rozhranie *GUI* v displeji *Nextion*. [11]

3.5. Zapojenie *Nextion* displeja s mikrokontrolérom

Nextion má zabudovaný *ARM* mikrokontrolér, ktorý ovláda displej a stará sa o generovanie tlačidiel, tvorbu textu, ukladanie obrázkov či zmenu pozadia.

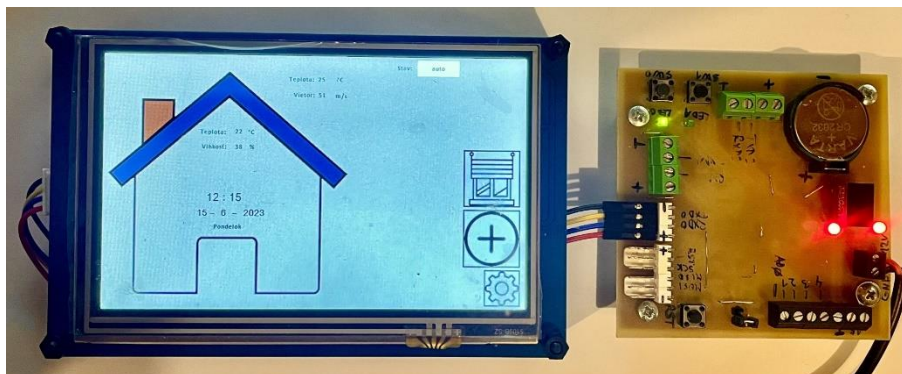
Nextion komunikuje s akýmkoľvek mikrokontrolérom pomocou sériovej komunikácie s prenosovou rýchlosťou 9600 bdt. Takže funguje s akoukoľvek vývojovou doskou, ktorá má sériové funkcie, či je to *Arduino*, *Raspberry Pi*, *ESP32*, *ESP8266*, *SMT 32* alebo v našom prípade *master* zariadenie s mikrokontrolérom *ATmega324PB* od *Atmelu*. [11]

Zapojenie displeja s mikrokontrolérom je veľmi jednoduché. Napájací 5 V vodič (červený) pripojíme na 5 V zdroj, *GND* vodič (čierny) pripojíme na zem a dátové vodiče prepojíme krížovo medzi sebou a to *RX* s *TX* a *TX* s *RX* podľa obrázka (obr. 18) nižšie.



Obr. 18 Zapojenie *HMI* displeja s *MCU* cez *UART*

Výsledné zapojenie naprogramovaného užívateľského rozhrania *GUI*, *HMI* displeja *Nextion* s mikrokontrolérom *ATmega324PB*, ktorý má funkciu *master*, je zobrazený na nasledujúcom obrázku (obr. 19). Ako môžeme vidieť, tak na komunikáciu medzi *master* zariadením a displejom nám stačia 4 vodiče.



Obr. 19 Zapojený *HMI* displej *Nextion* s *master* zariadením

3.6. Posielanie údajov z *Nextion* displeja do *master* zariadenia

Údaje prichádzajúce z displeja a údaje posielané do displeja musia dodržiavať určený formát. Zvyčajne je posielaných 5 - 9 bajtov dát vo voliteľnom kódovaní, predurčený typ je *ISO-8859-2*. Prvý bajt je riadiaci a upresňuje akú informáciu nesú ďalšie bajty. Všetky prichádzajúce dáta sú ukončené tromi znakmi *0xFF*. Celkovo výstup na komunikačnej linke *UART* vyzerá nasledovne:

00 00 00 FF FF FF
88 FF FF FF
A9 17 06 0F 0C 0F 01 00 F1

Obr. 20 Odosielanie údajov z displeja cez *UART*

Všetky kódy a príkazy, ktorými displej s *master* zariadením pracujú/komunikujú sú zobrazené v nasledujúcej tabuľke (tab. 1) a vysvetlené v texte pod tabuľkou.

Tab. 1 Komunikačné kódy a povelý displeja s *master* zariadením

1. bajt	2. bajt	3. bajt	4. bajt	5. bajt	6. bajt	7. bajt	8. bajt	9. bajt
0xA1	ID žalúzií	Pozícia	Natočenie	0	0	0	0	CHECKSUM
0xA3	ID žalúzií	Pozícia	0	0	0	0	0	CHECKSUM
0xA5	ID žalúzií	Minúty lokálnej doby	Minúty <i>master</i> doby	0	0	0	0	CHECKSUM
0xA7	ID žalúzií	Stav 0/1	0	0	0	0	0	CHECKSUM
0xA8	ID žalúzií 0xFF	0	0	0	0	0	0	CHECKSUM
0xA9	Rok	Mesiac	Deň	Hodina	Minúta	Deň	0	CHECKSUM
0xAB	Stav	Minúta	Hodina	Natočenie	Pozícia	ID žalúzií	Deň	CHECKSUM
0xAD	Stav	Minúta	Hodina	Natočenie	Pozícia	ID žalúzií	Deň	CHECKSUM
0xAF	Stav	Minúta	Hodina	Natočenie	Pozícia	ID žalúzií	Deň	CHECKSUM

Všetky kódy majú ošetrené posielanie dát cez *UART* zbernicu pomocou funkcie *CHECKSUM*. Táto funkcia posiela súčet všetkých hexadecimálnych čísel, ktoré sa nachádzajú v odosielačom polí/reťazci ako posledný bajt tohto poľa/reťazca. *Master* po prijatí všetkých bajtov/dát overí prijatú správu, tak že opätovne spočíta všetky bajty v polí/reťazci a porovná ho s posledným prijatým bajtom, čo je práve *checksum*.

Všetky hodnoty z displeja sú odosielané, cez UART zbernicu pomocou hexadecimálnych čísel podľa vyššie uvedenej tabuľky (tab. 1). Bližší popis kódov je vysvetlený v nasledujúcom texte.

Kód 0xA1 znamená priame nastavenie pozície a natočenia žalúzií. Po tomto plynie *master* doba.

Kód 0xA3 znamená priradenie maximálnej pozície zvoleným žalúziám.

Kód 0xA5 znamená priradenie časového limitu pre lokálnu dobu a *master* dobu.

Kód 0xA7 je určený pre informáciu stavu o umývaní okien.

Kód 0xA8 je určený pre zrušenie manuálneho režimu pre všetky žalúzie (0xFF).

Kód 0xA9 je určený pre priame nastavenie dátumu, času a dňa (Po - Ne) v *master* zariadení.

Kód 0xAB je určený pre prvý budík. Po kóde nasleduje stav budíka či je aktívny alebo neaktívny (1/0), potom nasleduje čas, natočenie a pozícia vybraných žalúzií, vybrané žalúzie, zvolené dni v týždni a *checksum*.

Kód 0xAD je určený pre druhý budík. Po kóde nasleduje stav budíka či je aktívny alebo neaktívny (1/0), potom nasleduje čas, natočenie a pozícia vybraných žalúzií, vybrané žalúzie, zvolené dni v týždni a *checksum*.

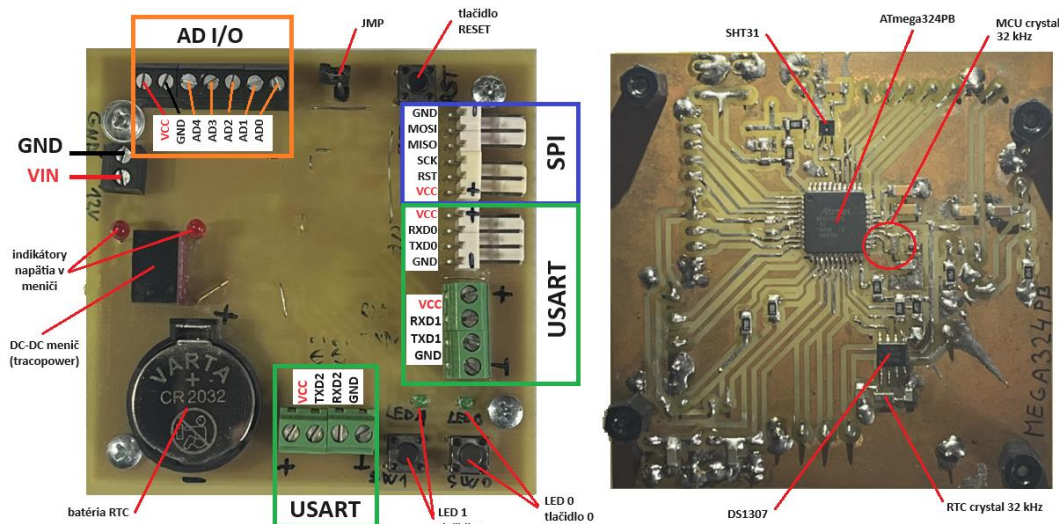
Kód 0xAF je určený pre tretí budík. Po kóde nasleduje stav budíka či je aktívny alebo neaktívny (1/0), potom nasleduje čas, natočenie a pozícia vybraných žalúzií, vybrané žalúzie, zvolené dni v týždni a *checksum*.

4. Master zariadenie pre ovládanie exteriérových žalúzií

Pre projekt ovládania exteriérových žalúzií bola vyvinutá nasledujúca vývojová doska *MASTER*, na ktorej sa nachádza čip *ATmega324PB*. Táto vývojová doska má zabudovaný *RTC* modul (*DS1307*) so zabudovanou batériou, snímač relatívnej vlhkosti/teploty (*SHT31*) s vlastným kryštálom, tlačidlá a *LED* indikátory, ktoré môžu byť použité pre rôzne funkcie pracujúce cez vstupno-výstupné piny (*GPIO*). Okrem toho má predpripravenú analógovo-digitálnu vstupno-výstupnú zbernicu pre budúce využitie priamo na doske.

Master zariadenie je možné programovať pomocou sériovej *SPI* zbernice, pričom *USART* zbernica je vyhradená pre komunikáciu s *Nextion* displejom a *slave* zariadeniami. Displej *HMI Nextion*, ktorý sme popísali v predchádzajúcej kapitole je pripojený, cez prvú *USART* zbernicu pomocou *RXD0* a *TXD0* pinov. Druhá *USART* zbernica je vyhradená pre *slave* zariadenia a tretia pre ostatné zariadenia, ako sú snímač vetra, osvetlenia alebo teploty a vlhkosti vonkajšieho prostredia.

Master zariadenie je napájané pomocou 12 V jednosmerného zdroja, ktoré sa mení na požadované 5 V jednosmerné napätie, s ktorým pracuje *master*, alebo koniec koncov aj *Nextion* displej a ostatné zariadenia.



Obr. 21 *Master* zariadenie s čipom *ATmega324PB*

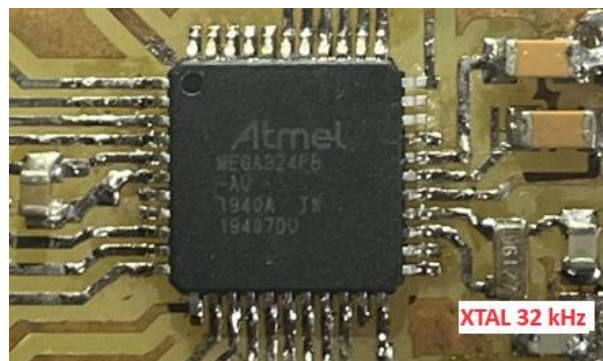
4.1. Mikrokontrolér *ATmega324PU*

V roku 1997 uviedla na trh firma *Atmel* nové 8-bitové mikrokontroléry rady AVR.

Mikrokontrolér *Atmel AVR* využíval architektúru *RISC (Reduced instruction set computing)*, čo prinieslo optimalizáciu a zlepšenie výkonu. [12]

ATmega324PB je 8-bitový mikrokontrolér s nízkou spotrebou energie založený na architektúre AVR od spoločnosti *Microchip Technology*, ktorý je používaný v rôznych aplikáciách, ako sú *embedded* systémy, robotika, *IoT* a iné. Mikrokontroléry *Atmel AVR* sú optimalizované pre programovanie v jazyku *C/C++*. [12]

V mikrokontroléri *ATmega324PB* sú tri druhy pamäte a to prvá pamäť *Flash*, ktorá má veľkosť 32 KB a slúži na ukladanie programu, táto pamäť je obnoviteľná. Druhá pamäť *SRAM* má veľkosť 2 KB a využíva sa na krátkodobé uloženie dát, ale pri prerušení napätia sa zmaže. Tretia pamäť *EEPROM* má veľkosť 1 KB a je určená pre jednorazové uloženie dát, ktoré sa zachovávajú bez potreby udržania napätia. [12][13]

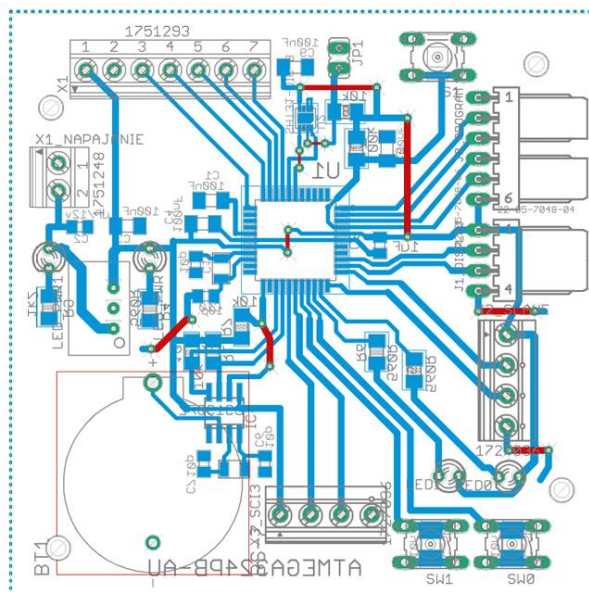


Obr. 22 Mikrokontrolér *ATmega324PB* z externým 32kHz kryštálom

Mikrokontrolér *ATmega324PB* taktiež ponúka širokú škálu periférií vrátane sériových komunikačných rozhraní (3x *USART*, 2x *SPI* a 2x *TXI/I2C*), 8 analógovo-digitálnych kanálov, dvoch 8-bitových a troch 16-bitových časovačov. *ATmega324PB* má flexibilné možnosti vstupno-výstupných pinov (39x *GPIO*). Okrem toho má bezpečnostné funkcie, ako sú *watchdog* časovače a ochranu pred zápisom. [12][13]

Mikrokontrolér *ATmega324PB* má viacero zdrojov hodinového signálu, vrátane interného oscilátora s nastaviteľnou frekvenciou, čo umožňuje mikrokontroléru fungovať aj bez externého hodinového signálu. Mikrokontrolér môže prijať aj externý hodinový signál z externého oscilátora alebo kryštálu ako je zobrazené na obrázku vyššie (obr. 22), kde je pripojený 32kHz kryštál. [13]

Zapojenie je realizované na obojstrannom *DPS* (doska plošných spojov) 73,8 x 73,5mm. Na zapojenie boli použité klasické súčiastky podľa zoznamu (Tab. 2).



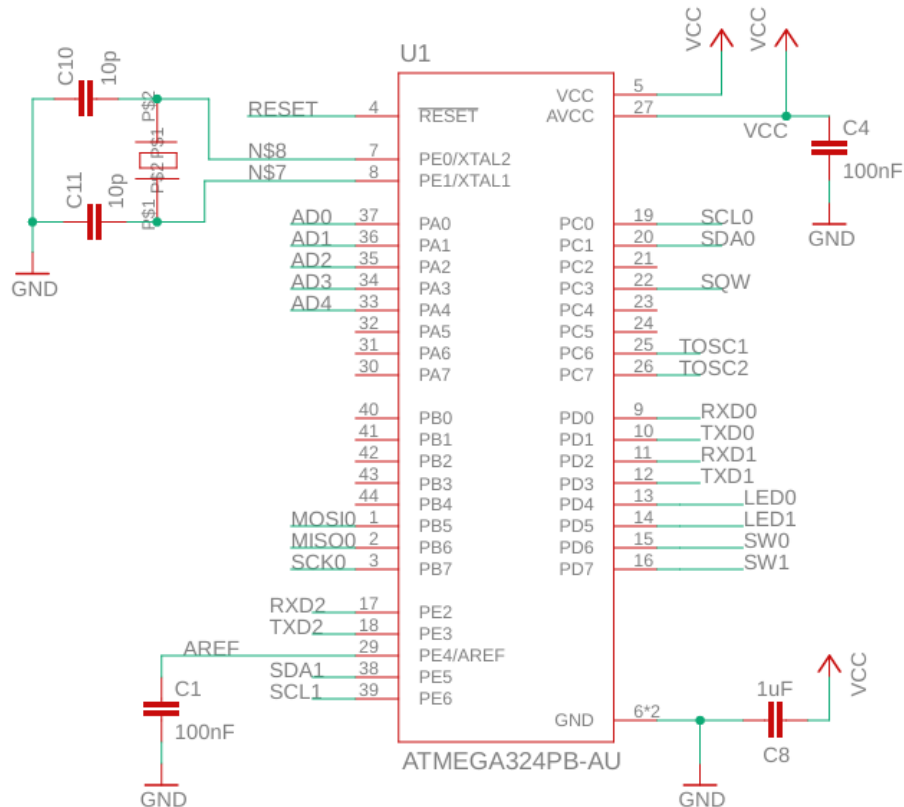
Obr. 23 Úplne zapojenie *DPS* master zariadenia

Tab. 2 Zoznam súčiastok *DPS* master zariadenia

Batériový modul	BAT_BS-7
C_SW0, C_SW1	10 nF
C1, C3, C4, C5, C9	100 nF
C2, C8	1 μ F
C6, C7, C10, C11	10 pF
IC1 (RTC)	DS1307Z
LED_PWR, LED_PWR1	Červená dióda
LED0, LED1	Zelená dióda
R1, R2, R7, R8	10 k Ω
R3	100 k Ω
R4, R5, R6	560 Ω
R9	1,2 k Ω
SW0, SW1 (tlačidlá)	
DC-DC menič (Tracopower)	TSR 1-2450
2x kryštál	IL3X2 32,768 kHz
U1 (MCU)	ATmega324PB
U2 (snímač teploty/vlhkosti)	SHT31

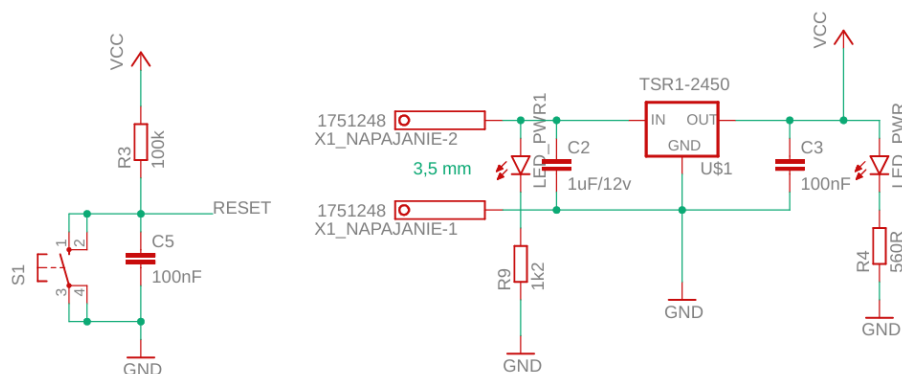
4.1.1. Zapojenie mikrokontroléra *ATmega324PB*

Nasledujúce schématické zobrazenie poukazuje na zapojenie obvodu mikrokontroléra *ATmega324PB* a externého kryštálu 32,768 kHz so zapojenými kondenzátormi.



Obr. 24 Elektrická schéma obvodu mikrokontroléra s obvodom externého kryštálu

V resetovacom a napájacom obvode sú okrem kondenzátorov pripojené v obvode aj rezistory. Napájací obvod je zložený z DC-DC meniča *TSR 1-2450* (jednosmerný spínací regulátor) od výrobcu *tracopower*. Napájací obvod mení vstupné 6,5 V až 36 V jednosmerné napätie na výstupné 5 V jednosmerné napätie a prúd 1 A, s ktorým ďalej pracuje *master* zariadenie a *Nextion* displej. Obvod obsahuje kontrolné LED diódy pre indikáciu vstupného a výstupného napätia.



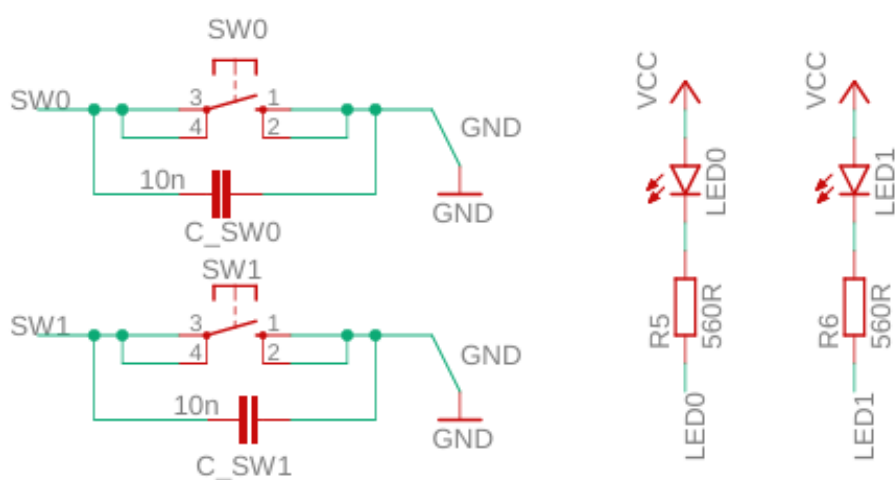
Obr. 25 Elektrická schéma resetovacieho (vľavo) a napájacieho obvodu (vpravo)

4.1.2. Zapojenie tlačidiel a LED diód

Na nasledujúcom obrázku je zobrazená elektrická schéma pripojenia tlačidiel a LED diód plošného spoja *DPS*. Tlačidlá *SW0* a *SW1* sú prepojené s kondenzátormi *C_SW0* a *C_SW1* voči *GND*. LED diódy sú zapojené s rezistormi v sérii podľa ohmového zákona (vzorec 1), tak aby sa dosiahol bezpečný pracovný prúd pre 5 V vstupné napätie zelenej LED diódy a tým ochránil LED pred poškodením. Odpor pre LED diódy je 560 Ω.

$$I = \frac{U}{R} = \frac{5}{560} = 0,0089 \text{ A} \quad [1]$$

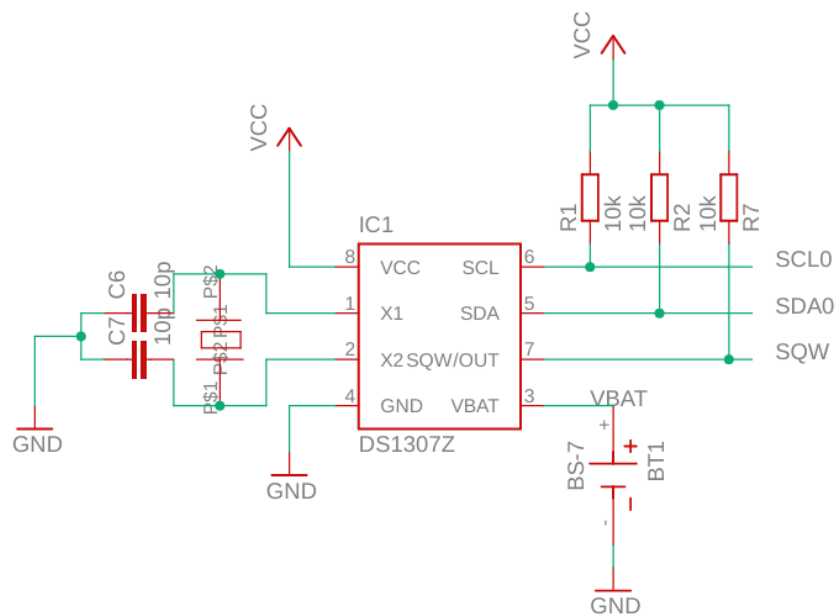
LED diódami bude pretekať 8,9 mA prúd.



Obr. 26 Elektrická schéma obvodov pre tlačidlá a LED diódy

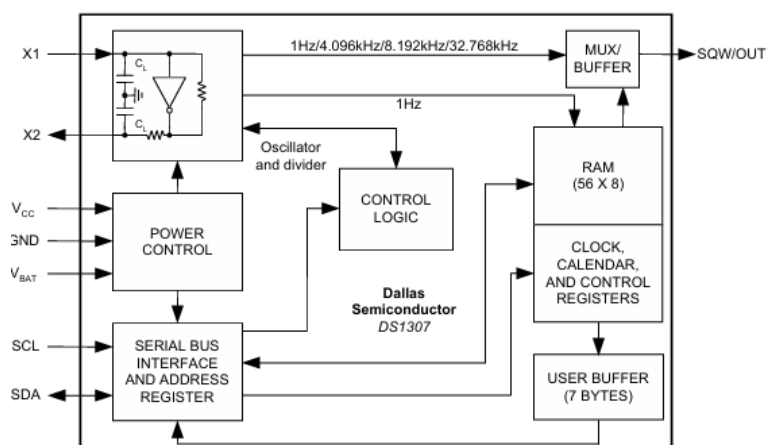
4.1.3. Zapojenie senzora reálneho času a dátumu DS1307

Na nasledujúcom obrázku je zobrazený elektrický obvod pre RTC modul, snímač reálneho dátumu a času DS1307. V obvode je zapojený 32,768 kHz kryštál s filtrovaním pomocou kondenzátorov. Jednotlivé komunikačné piny TWI/I2C zbernice a to SCL a SDA sú pripojene s rezistormi na VCC ako *pull-up*. Okrem toho je, ako *pull-up* pripojený aj SQW (Square Wave) pin. Na pin VBAT je pripojená externá batéria, ktorá slúži pre udržanie napätia RTC modulu a tým je ošetrená strata údajov RTC po vypnutí *master* zariadenia.



Obr. 27 Elektrická schéma obvodu RTC modulu DS1307

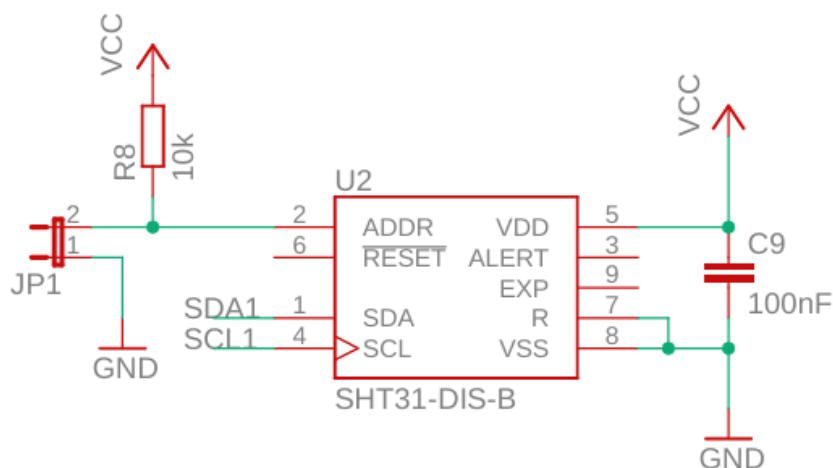
Na nasledujúcom obrázku je zobrazená funkčná bloková schéma DS1307.



Obr. 28 Funkčná bloková schéma DS1307 [14]

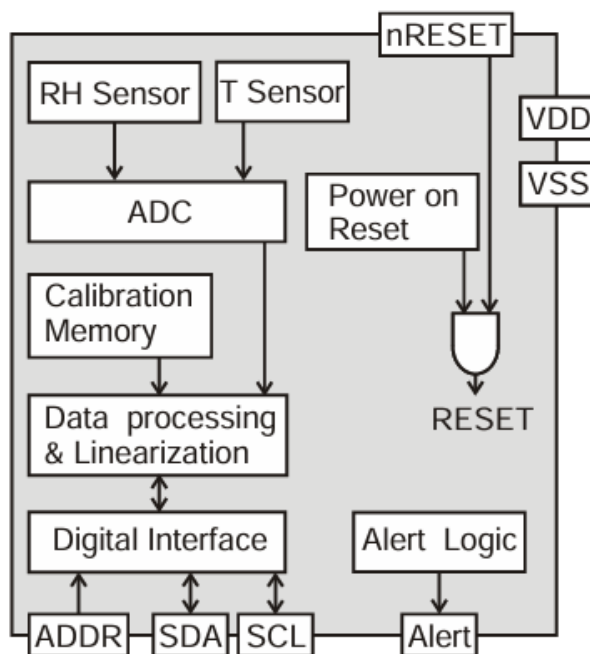
4.1.4. Zapojenie senzora teploty a vlhkosti SHT31

Na nasledujúcom obrázku je zobrazený elektrický obvod pre snímač teploty a relatívnej vlhkosti SHT31. Komunikačná zbernica TWI/I2C (SDA, SCL piny) je ošetrená *pull-up* rezistormi v predchádzajúcom obvode (obr. 27), preto v tomto obvode nie sú zahrnuté. Pin ADDR je pripojený sériovo s rezistorom na VCC a pomocou prepájača JP1 sa môže spojiť so zemou GND.



Obr. 29 Elektrická schéma snímača relatívnej teploty a vlhkosti SHT31

Na nasledujúcom obrázku je zobrazená funkčná bloková schéma SHT31.



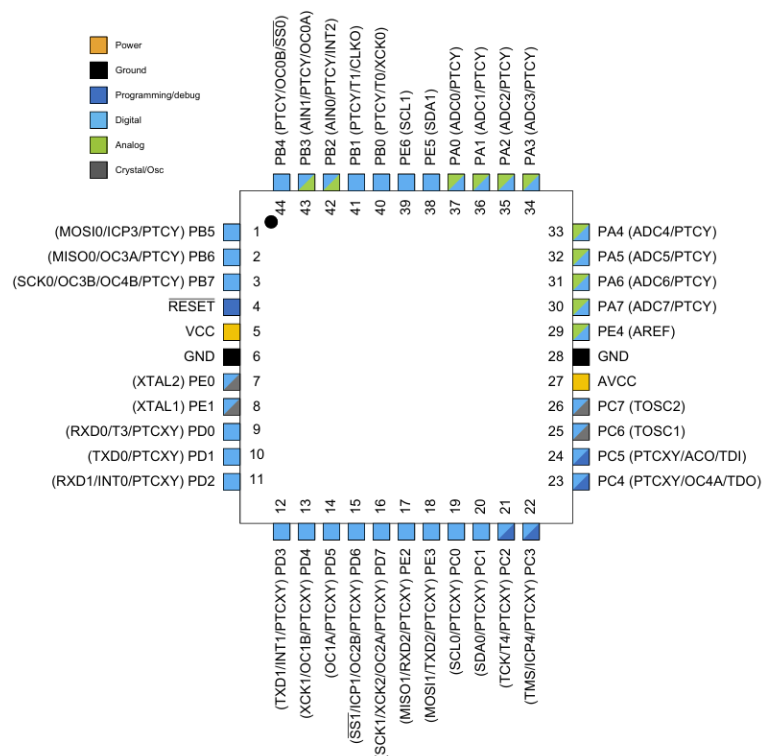
Obr. 30 Funkčná bloková schéma SHT31 [15]

4.2. Programovanie *master* zariadenia

Pri programovaní *master* zariadenia boli dodržané nasledujúce pravidlá programovania. Ako prvé bolo nevyhnutné dohľadať a stiahnuť konkrétny *datasheet* mikrokontroléra *ATmega324PB*, s ktorým sa ďalej pracovalo. Nachádzali sa tam všetky potrebné informácie pre prácu s mikrokontrolérom. Program bol písaný pomocou *Microchip studio*. Pri celom rozsahu písania kódu boli použité komentáre pre lepšiu orientáciu v programe. Dodržali sa pravidlá formátovania použitím štruktúrovaného zápisu zátvoriek, správneho odsadzovania, logiky písania kódu a použitím funkcií a hlavičkových súborov pre lepšiu prehľadnosť programu.

Všetky definície funkcií, konštánt a tiež určitých premenných, s ktorými pracuje *master* zariadenie sú umiestnené a logicky rozdelené v hlavičkových súboroch. Ku hlavičkovým súborom boli jednotlivo vytvorené samostatné „c“ zložky, kde sú inicializované a naprogramované všetky funkcie. Týmto je zabezpečená lepšia orientácia a prehľadnosť celého projektu.

Pri programovaní mikrokontrolérov je nevyhnutné porozumieť a naučiť sa pracovať s registrami konkrétného mikrokontroléra. Práve vďaka správne nastaveniu registrov dokážeme ovládať komunikáciu mikrokontroléra s jeho perifériami. Medzi periférie mikrokontroléra patria *GPIO* porty, na ktorých sú *SPI*, *USART*, *I2C* zbernice, *A/D* prevodníky, čítače-časovače a ďalšie.



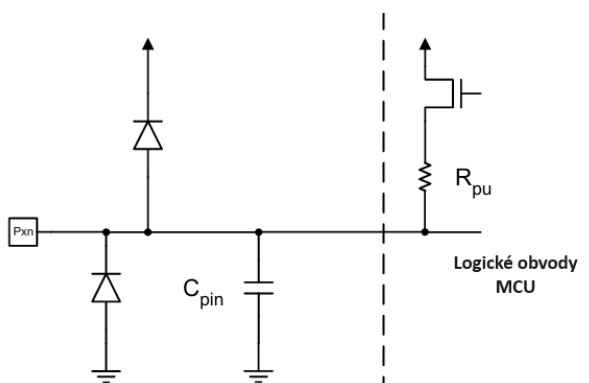
Obr. 31 Vstupno-výstupné piny mikrokontroléra *ATmega324PB*

4.2.1. Vstupno-výstupné GPIO porty a CPU hodiny

V mikrokontroléri *ATmega324PB* sa rozlišujú pracovné, príznakové a nastavovacie registre. *Atmega324PB* disponuje so 44 pracovnými registrami, ktoré sú 8 bitové a slúžia na ukladanie premenných počas behu programu. Tieto registre manažuje kompilátor programovacieho jazyka C. Príznakové registre informujú o stave portov v perifériách. Pomocou nastavovacích registrov je možné nastavovať jednotlivé piny a parametre mikrokontroléra. [12]

Každý *GPIO* port má 8 bitov, čo predstavuje jeden bajt a tie sú vyvedené ako piny čipu mikrokontroléra. Porty sú označené podľa abecedy a to *PORTA*, *PORTB*, *PORTC* a *PORTD*, kde každý bit/pin v porte je označený ako *PORTA0* až *PORTA7*, *PORTB0* až *PORTB7*, *PORTC0* až *PORTC7* a *PORTD0* až *PORTD7* (tiež je možné sa stretnúť s označením *PA0* až *PA7*, *PB0* až *PB7*, *PC0* až *PC7* a *PD0* až *PD7*). [13]

Pre každý port je možné nastavovať všetky piny, buď ako vstup alebo výstup. Podľa nastavenia registrov určité piny v porte môžu mať funkcionality ako *UART*, *SPI*, *TWI/I2C* alebo externý kryštál. Všetky piny sú chránené pomocou diód priamo v čipe. Piny by sa nemali preťažovať väčším prúdom ako 40 mA. Mikrokontrolér *ATmega324PB* pracuje s logickou 0, čo znamená, že pri aktivácii *pull-up* rezistora sa rezistor pripája na GND. V projekte sú aktivované interné *pull-up* rezistory mikrokontroléra. [13]



Obr. 32 Vnútna štruktúra vstupno-výstupných pinov mikrokontroléra

Nízkofrekvenčný oscilátor je optimalizovaný pre použitie s kryštálom hodín 32,768 kHz. V predvolenom nastavení poskytuje interný RC oscilátor hodiny 8 MHz, ktoré sú závislé od napätia a teploty. [13]

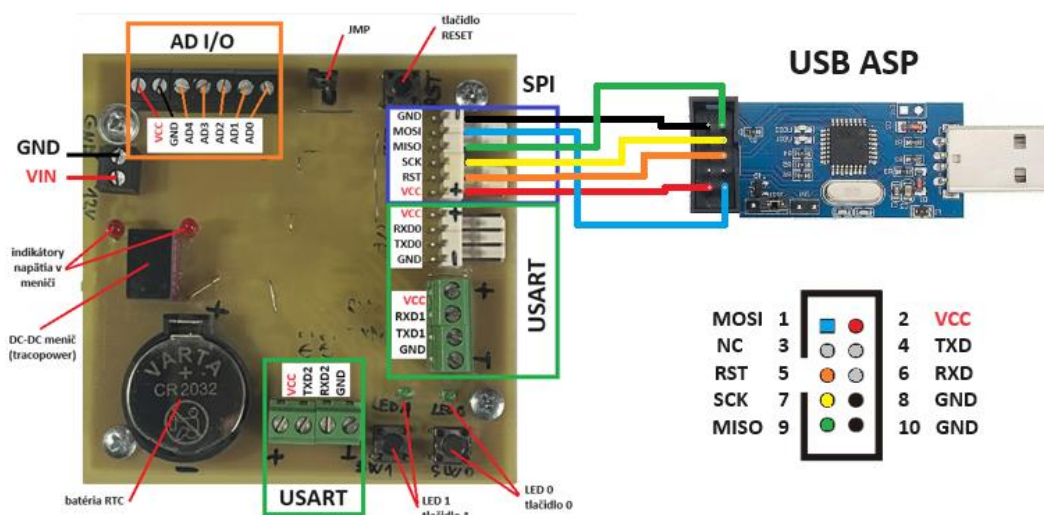
4.2.2. Programovanie mikrokontroléra pomocou SPI komunikácie

Na programovanie mikrokontroléra sa používal najpoužívanejší spôsob programovania, cez SPI zbernicu. AVR mikrokontrolér *ATmega324PB* bol programovaný pomocou *ISP (In System Programming)*, cez sériové rozhranie pomocou prevodníka *USB-ASP* pripojeného k počítaču v *API* od *Microchip Studio*.

Desať pinový konektor pre programovanie pomocou SPI rozhrania zobrazený na obrázku nižšie (obr. 33) je štandardom *ISP* programovania. Na programovanie sa využili štyri dátové vodiče *MOSI (Master Out Slave In)*, *MISO (Master In Slave Out)*, *SCK (Serial Clock)* a *RESET*. Okrem dátových vodičov sa využili aj vodiče pre napájanie *VCC (Voltage Common Collector)* a spoločný uzemňovací vodič *GND (Ground)*.

Po prepojení vodičov bolo nevyhnutné nainštalovať ovládač *USB-ASP* prevodníka a nakonfigurovať pomocou *AVRDUDE ISP* rozhranie v *Microchip Studio* editore. Nastavil sa pre programovanie čip *ATmega324PB* od *Atmel-u*. Po kompletnom nastavení bolo možné napaľovať program, cez *SPI* rozhranie do mikrokontroléra *ATmega324PB* (*master* zariadenie).

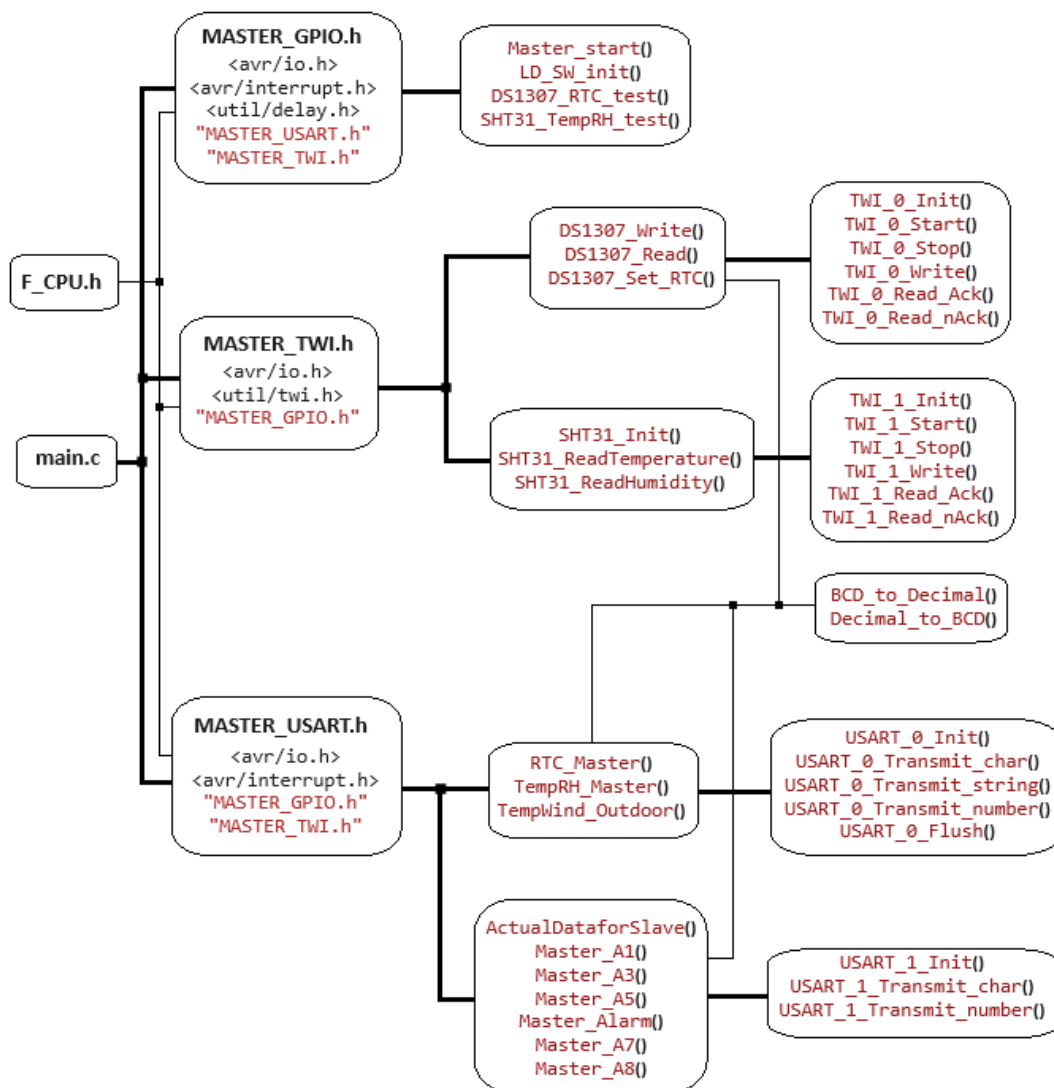
Na nasledujúcom obrázku je zobrazené zapojenie *USB-ASP* prevodníka s *master* zariadením, cez *SPI* zbernicu. Toto zapojenie sa využilo len pre nahrávanie zdrojového kódu do mikrokontroléra.



Obr. 33 Zapojenie SPI prevodníka *USB-ASP* s *MCU*

4.3. Štruktúra zdrojového kódu *master* zariadenia

Na nasledujúcom obrázku (obr. 34) je zobrazená štruktúra zdrojového kódu *master* zariadenia pre ovládanie exteriérových žalúzií. Projekt je zložený zo štyroch hlavných knižníc, ktoré obsahujú definície, inicializácie a naprogramované funkcie potrebné pre celý projekt. Program obsahuje naprogramované funkcie pre komunikačné (I2C, USART, GPIO) zbernice, s ktorými pracujú snímače SHT31 a DS1307 a taktiež Nextion displej a *slave* zariadenia. Pre overovanie snímačov a komunikácií sú v projekte tiež testovacie funkcie.



Obr. 34 Štruktúra zdrojového kódu pre *master* zariadenie

V projekte je predpripravená funkcia pre snímanie teploty, vetra a osvetlenia z vonkajšieho prostredia. Nachádza sa v knižnici „MASTER_USART.h“ pod označením „TempWind_Outdoor()“.

4.3.1. Program pre *master* vstupno-výstupné GPIO piny

V hlavičkovom súbore „*MASTER_GPIO.h*“ sa nachádzajú deklarované funkcie pre prácu s *GPIO*, ktoré sú naprogramované v samostatných „.c“ súboroch obsahujúce zdrojové kódy. Tieto funkcie sú programované pomocou registrov, ktoré využívajú logické a bitové operácie. V knižnici sú tiež definované a nastavené jednotlivé porty pre tlačidlá a *LED* diódy, ktoré využívajú ostatné funkcie v programe. Jednotlivé zdrojové kódy sú vysvetlené nižšie v texte.

Definícia *LED* diód v zapnutom stave (log. 0) na porte *PD4* a *PD5* využíva nasledujúce kódy:

```
#define LED0_ON      (PORTD &= ~(1<<PORTD4))
#define LED1_ON      (PORTD &= ~(1<<PORTD5))
```

Definícia *LED* diód vo vypnutom stave (log. 1) na porte *PD4* a *PD5* využíva:

```
#define LED0_OFF     (PORTD |= (1<<PORTD4))
#define LED1_OFF     (PORTD |= (1<<PORTD5))
```

Definícia *LED* diód, ktorá vyčítava hodnoty pinov a stavu na porte (*PD4* a *PD5*) je:

```
#define LED0         (PIND & (1<<PORTD4))
#define LED1         (PIND & (1<<PORTD5))
```

Definícia tlačidiel vyčítavajúce hodnoty pinov na porte *PD6* (*SW0*) a *PD7* (*SW1*) využíva:

```
#define SW0          (PIND & (1<<PORTD6))
#define SW1          (PIND & (1<<PORTD7))
```

Funkcia „*Master_start*“ je určená pre indikáciu začiatku behu programu. Je zadefinovaná v *main.c* súbore v hlavnej funkcii *main()*, pred nekonečným cyklom *while(1)*. Táto funkcia 3 krát preblikne s oboma *LED* diódami súčasne. Funkcia „*LD_SW_init*“ nastavuje pre tlačidlá hodnotu vstupu (logická 0) v registri *DDR* a *pull-up* rezistory (logická 1) na porte:

```
DDRD = (1<<DDD4)|(1<<DDD5);
DDRD &= ~((1<<PORTD6)|(1<<PORTD7));
PORTD = (1<<PORTD6)|(1<<PORTD7);
```

Funkcia „*DS1307_RTC_test*“ slúži pre testovanie reálneho dátumu a času *RTC* modulu (*DS1307*), po sériovej komunikácii s *Nextion* displejom. Funkcia „*SHT31_TempRH_test*“ slúži pre testovanie relatívnej teploty a vlhkosti snímača *SHT31* po sériovej komunikácii s *Nextion* displejom. Tieto testovacie funkcie sú len orientačné a zakomentované (v projekte pre žalúzie sú nevyužitú).

4.3.2. Program pre *master* sériovú TWI/I2C komunikáciu

V hlavičkovom súbore „*MASTER_TWI.h*“ sa nachádzajú deklarované funkcie pre prácu s komunikačnou zbernicou *TWI/I2C*, ktoré sú naprogramované v samostatných „.c“ súboroch obsahujúce zdrojové kódy. Tieto funkcie sú programované pomocou registrov, ktoré využívajú logické a bitové operácie. Prvú zbernicu (*TWI 0*) využíva snímač *DS1307* a druhú (*TWI 1*) využíva snímač *SHT31*. Ako prvé je v knižnici definovaná premenná prenosovej rýchlosti (*SCL_CLOCK*) na hodnotu 100 KHz, pričom maximálna hodnota prenosovej funkcie môže byť nastavená do 400 KHz. V zdrojovom súbore „*MASTER_TWI.c*“ sú definované konkrétne adresy snímačov pomocou *datasheet-u*. Pre *RTC* snímač (*DS1307*) platí:

```
#define DS1307_WRITE  0xD0    - Adresa pre zápis hodnoty.
#define DS1307_READ   0xD1    - Adresa pre čítanie hodnoty.
```

Pre snímač teploty a vlhkosti (*SHT31*) platia nasledujúce definície:

```
#define SHT31_ADDRESS  0x44    - Adresa pre čítanie hodnoty.
#define SHT31_HIGH_TEMP 0x2C    - Nastavovacia adresa teploty MSB (najvyšší bit).
#define SHT31_LOW_TEMP  0x06    - Nastavovacia adresa teploty LSB (najnižší bit).
#define SHT31_HIGH_HUM  0x24    - Nastavovacia adresa vlhkosti MSB (najvyšší bit).
#define SHT31_LOW_HUM   0x00    - Nastavovacia adresa vlhkosti LSB (najnižší bit).
```

Nastavenie *TWI/I2C* komunikácie vyžaduje inicializáciu, ktorá je naprogramovaná funkciou „*TWI_0_Init*“ pre snímač *DS1307* a „*TWI_1_Init*“ pre snímač *SHT31*. Nastavenie rýchlosti prenosu dát na hodnotu 100 KHz je dosiahnuté pomocou „*bit-rate*“ registra *TWBR*, ktorý je vyjadrený podľa vzorca z *datasheet-u* mikrokontroléra:

$$TWBR = \frac{\frac{F_{CPU}}{SCL_CLOCK} - 16}{2} \quad [2]$$

Nastavenie inicializácie je dokončené pomocou stavového registra *TWSR* s registrom predeličky (*TWPS*) a *pull-up* rezistormi, ktoré využívajú *TWI/I2C* zbernicu *SDA (PC0)*, *SCL (PC1)*:

```
TWSR0 &= ~(1 << TWPS0);
TWSR0 &= ~(1 << TWPS1);
DDRC &= ~((1<<PORTC0)|(1<<PORTC1));
PORTC = (1<<PORTC0)|(1<<PORTC1);
```

Funkcia „*TWI_0_Start*“ pre snímač *DS1307* a „*TWI_1_Start*“ pre snímač *SHT31* slúži na odoslanie štart podmienky v komunikácií. Funkcia „*TWI_0_Stop*“ a „*TWI_1_Stop*“ slúži na odoslanie stop podmienky v komunikácií.

Funkcia „*TWI_0_Write*“ a „*TWI_1_Write*“ slúži na vysielanie dátového bajtu na zbernicu pomocou dátového registra *TWDR*. Funkcia „*TWI_0_Read_Ack*“ a „*TWI_1_Read_Ack*“ slúži na príjem dátového bajtu zo zbernice pomocou registra *TWCR*, po ktorom nasleduje ďalší dátový bajt do dátového registra *TWDR*. Funkcia „*TWI_0_Read_nAck*“ a „*TWI_1_Read_nAck*“ je podobná predchádzajúcej funkcii s tým rozdielom, že vráti posledný prijatý bajt, po ktorom nasleduje stop bit.

Funkcia „*BCD_to_Decimal*“ slúži na prevod z binárneho čísla (*bcd*) do desiatkového celého čísla: $((bcd \gg 4) * 10) + (bcd \& 0x0F)$. Funkcia „*Decimal_to_BCD*“ slúži na prevod z celého desiatkového čísla (*decimal*) na binárne číslo postupne získaním desiatok (*tens*): $tens = decimal / 10$ a jednotiek (*units*): $units = decimal \% 10$. Spojením desiatok a jednotiek vznikne binárne číslo v BCD formáte: $(tens \ll 4) | units$.

Funkcia „*DS1307_Write*“ slúži pre zápis dátumu, dňa a času pomocou *TWI_0* zbernice. Funkcia „*DS1307_Read*“ slúži pre čítanie dátumu, dňa a času z *RTC* modulu *DS1307* pomocou *TWI_0* zbernice. Funkcia „*DS1307_Set_RTC*“ slúži na priame nastavenie dátumu, dňa a času do *RTC* modulu *DS1307* pomocou funkcií „*DS1307_Write*“ a „*Decimal_to_BCD*“.

V nasledujúcej tabuľke sú zobrazené jednotlivé adresy s umiestnením bitov pre sekundy (*0x00*), minúty (*0x01*), hodiny (*0x02*), dni v týždni pondelok až nedeľa (*0x03*), dátum (*0x04*), mesiac (*0x05*) a rok (*0x06*). V tabuľke je zobrazený aj ich rozsah (*RANGE*).

Tab. 3 *RTC* adresy s umiestnením bitov pre hodnoty dátumu a času v *DS1307*

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00H	CH	10 Seconds			Seconds				Seconds	00–59
01H	0	10 Minutes			Minutes				Minutes	00–59
02H	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03H	0	0	0	0	0	DAY			Day	01–07
04H	0	0	10 Date		Date				Date	01–31
05H	0	0	0	10 Month	Month				Month	01–12
06H	10 Year				Year				Year	00–99
07H	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08H-3FH									RAM 56 x 8	00H–FFH

Funkcia „*SHT31_Init*“ slúži pre inicializáciu *TWI_1* zbernice pomocou „*TWI_1_Init*“ funkcie. Funkcia „*SHT31_ReadTemperature*“ slúži pre čítanie teploty zo snímača *SHT31*. Využíva predchádzajúce funkcie pre komunikáciu *TWI_1* zbernice (začiatok, zápis, čítanie a ukončenie).

Nižšie je zobrazený začiatok komunikácie a postup adres snímača s príkazom na meranie teploty:

```
TWI_1_Start();
TWI_1_Write(SHT31_ADDRESS << 1);
TWI_1_Write(SHT31_HIGH_TEMP);
TWI_1_Write(SHT31_LOW_TEMP);
```

Čítanie údajov teploty s ukončením *TWI_1* komunikácie:

```
TWI_1_Start();
TWI_1_Write((SHT31_ADDRESS << 1) | 0x01);
temperature_raw = TWI_1_Read_nAck() << 8;
temperature_raw |= TWI_1_Read_nAck();
TWI_1_Stop();
```

, kde premenná „*temperature_raw*“ predstavuje výslednú hodnotu teploty zo snímača, ktorá je prepočítaná podľa vzorca z *datasheet-u*:

$$Teplota = -45 + 175 \frac{Teplota\ snímača}{2^{16} - 1} \quad [3]$$

Funkcia „*SHT31_ReadHumidity*“ slúži pre čítanie relatívnej vlhkosti (*RH*) zo snímača *SHT31*. Taktiež využíva predchádzajúce funkcie pre komunikáciu *TWI_1* zbernice (začiatok, zápis, čítanie a ukončenie). Nižšie je zobrazený začiatok komunikácie a postup adres snímača s príkazom na meranie relatívnej vlhkosti:

```
TWI_1_Start();
TWI_1_Write(SHT31_ADDRESS << 1);
TWI_1_Write(SHT31_HIGH_HUM);
TWI_1_Write(SHT31_LOW_HUM);
```

Čítanie údajov relatívnej vlhkosti s ukončením *TWI_1* komunikácie:

```
TWI_1_Start();
TWI_1_Write((SHT31_ADDRESS << 1) | 0x01);
humidity_raw = TWI_1_Read_nAck() << 8;
humidity_raw |= TWI_1_Read_nAck();
TWI_1_Stop();
```

, kde premenná „*humidity_raw*“ predstavuje výslednú hodnotu relatívnej vlhkosti zo snímača, ktorá je prepočítaná podľa vzorca z *datasheet-u*:

$$Relatívna\ vlhkosť = 100 \frac{Vlhkosť\ snímača}{2^{16} - 1} \quad [4]$$

4.3.3. Program pre *master* sériovú USART komunikáciu

V hlavičkovom súbore „MASTER_USART.h“ sa nachádzajú deklarované funkcie pre prácu s komunikačnou zbernicou USART, ktoré sú naprogramované v samostatných „.c“ súboroch obsahujúce zdrojové kódy. Tieto funkcie sú programované pomocou registrov, ktoré využívajú logické a bitové operácie. Prvú zbernicu (USART 0) využíva displej *Nextion* a druhú zbernicu (USART 1) využívajú *slave* zariadenia pre komunikáciu. Ako prvé je v knižnici definovaná premenná prenosovej rýchlosti *baud-rate* (BAUD) na hodnotu 9600 bdt. V zdrojovom súbore „MASTER_USART.h“ je definované pole (UART_0_DATA), pre prichádzajúce správy z *Nextion* displeja. Veľkosť poľa (ARRAY_SIZE) je definovaná s hodnotou 9.

Nastavenie USART komunikácie vyžaduje inicializáciu, ktorá je naprogramovaná funkciou „USART_0_Init“ pre komunikáciu s *Nextion* displejom a „USART_1_Init“ pre komunikáciu so *slave* zariadeniami. Nastavenie rýchlosti prenosu dát na hodnotu 9600 bdt je dosiahnuté pomocou „baud-rate“ registrov UBRRH a UBRRL, ktoré sú vyjadrené podľa vzorca z *datasheet-u* mikrokontroléra:

$$UBRR = \frac{F_{CPU}}{16 \cdot BAUD} - 1 \quad [5]$$

Register UBRRH = (UBRR >> 8) a register UBRRL = UBRR. Nastavenie inicializácie je dokončené pomocou riadiacich a stavových registrov UCSRA, UCSRB, a UCSRC:

UCSR0B = (1<<RXEN);	- Povolenie prijímania RX.
UCSR0B = (1<<TXEN);	- Povolenie vysielania TX.
UCSR0B = (1<<RXCIE);	- Povolenie prerušenia od príjmu RX.
UCSR0C = (1<<USBS) (1<<UCSZ1) (1<<UCSZ0);	- Nastavenie frame formátu.

Funkcia „USART_0_Transmit_char“ pre komunikáciu s *Nextion* displejom a funkcia „USART_1_Transmit_char“ pre komunikáciu so *slave* zariadeniami slúži na odoslanie textového znaku (vo formáte ASCII) po sériovej komunikácii. Dáta sa vkladajú do UDR0 a UDR1 registra. Funkcia „USART_0_Transmit_string“ slúži na odoslanie textového reťazca po sériovej komunikácii využívajúc funkciu „USART_0_Transmit_char(*str++)“;“. Funkcia „USART_0_Transmit_number“ a „USART_1_Transmit_number“ slúži na odoslanie desiatkového čísla (vo formáte ASCII) po sériovej komunikácii. Je navrhnutá len do troch cifier.

Ukážka zdrojového kódu je zobrazená nižšie na nasledujúcom obrázku (obr. 35).

```

if(num < 0){
    USART_0_Transmit_char('-');
    num *= -1;
}
// Prepocet zaporneho cisla na kladne cislo
// Odoslanie zaporneho znamienka

if(num >= 0 && num < 10){
    USART_0_Transmit_char(num + 0x30);
} // Jednociferné kladné číslo [x]
} else if(num >= 10 && num < 100){
    USART_0_Transmit_char((num / 10) + 0x30);
    USART_0_Transmit_char((num % 10) + 0x30);
} // Dvojčiferné kladné číslo [xx]
} else if(num >= 100 && num < 1000){
    USART_0_Transmit_char((num / 100) + 0x30);
    USART_0_Transmit_char(((num % 100) / 10) + 0x30);
    USART_0_Transmit_char((num % 10) + 0x30);
} // Trojčiferné kladné číslo [xxx]
}

```

Obr. 35 Zdrojový kód funkcií pre odoslanie čísla po sériovej USART komunikácii

Funkcia „*USART_0_Flush*“ slúži na vyprázdnenie prijímaných údajov (*Flushing the Receive Buffer*) zbieraním dát z údajového registra (*UDR0*) do nepoužitej premennej (*dummy*), ktorá je typu „*void*“.

Funkcia „*RTC_Master*“ slúži na odosielanie aktuálneho dátumu a času čítaním RTC modulu z *DS1307* priamo do *Nextion* displeja, cez *USART_0* zbernicu. Odosielané sú hodnoty dátumu, dňa a času, ktoré *Nextion* zariadenie prijíma ako premenné. Všetky parametre odosielané po sériovej zbernici sú ukončované odoslaním troch hexadecimálnych čísel *0xFF*.

Funkcia „*TempRH_Master*“ slúži na odoslanie vnútornej teploty a vlhkosti čítaním zo snímača *SHT31* priamo do *Nextion* displeja, cez *USART_0* zbernicu. Princíp odosielania hodnôt je rovnaký ako v predchádzajúcom vysvetlení.

Funkcia „*TempWind_Outdoor*“ je akosi predprípravou na odosielanie vonkajšej teploty a rýchlosti vetra do displeja. Sú predpripravené premenné a rozbehnutá komunikácia s displejom. Táto funkcia nie je dokončená pretože nebola predmetom zadania. Je však potrebné ju dokončiť.

4.3.4. Program pre *slave* sériovú *USART* komunikáciu

Nasledujúce funkcie spadajú pod kategóriu, ktoré sú určené pre *slave* zariadenia.

Funkcia „*ActualDataforSlave*“ slúži na odoslanie vnútornej a vonkajšej teploty, osvetlenia a vietra všetkým *slave* zariadeniam, cez *USART_1* zbernicu. Vnútna teplota je čítaná zo snímača *SHT31*, vonkajšia teplota, vietor a osvetlenia sú fiktívne a treba ich dokončiť pomocou funkcie „*TempWind_Outdoor*“. Odosielané sú premenné po sériovej komunikácii podľa tabuľky povelov pre *slave* zariadenia, ktorá je zobrazená nižšie (tab. 4).

Funkcia „*Master_A1*“ slúži pre priame zadanie povelu nastavenia žalúzií. Tento príkaz bol vydaný *master* zariadením priamo z displeja, cez hexadecimálny kód *0xA1*.

Funkcia „*Master_A3*“ slúži pre priradenie maximálnej pozície zvolených žalúzií v displeji. Tento príkaz bol taktiež vydaný *master* zariadením priamo z displeja, cez hexadecimálny kód *0xA3*.

Funkcia „*Master_A5*“ slúži pre priradenie lokálnej a *master* doby zvoleným žalúziám. Tento príkaz bol taktiež vydaný *master* zariadením priamo z displeja, cez hexadecimálny kód *0xA5*.

Funkcia „*Master_A7*“ slúži pre informáciu *slave* zariadeniam o umývaní vybraných okien. Tento príkaz bol taktiež vydaný *master* zariadením priamo z displeja, cez hexadecimálny kód *0xA7*.

Funkcia „*Master_A8*“ slúži pre informáciu *slave* zariadeniam o zrušení manuálneho režimu. Tento príkaz bol taktiež vydaný *master* zariadením priamo z displeja, cez hexadecimálny kód *0xA8*.

Master všetky tieto situácie spracuje a odošle premenné (povely zobrazené v tabuľke 4) *slave* zariadeniam po sériovej *USART_1* zbernici.

Funkcia „*Master_Alarm*“ je špeciálna funkcia, ktorá ovláda všetky tri budíky nastavené užívateľom priamo z displeja. Ako prvé sú nastavené premenné pre overenie aktuálneho času:

Minúty: `minutes = BCD_to_Decimal(DS1307_Read(0x01));`

Hodiny: `hours = BCD_to_Decimal(DS1307_Read(0x02));`

Deň v týždni: `day_n = BCD_to_Decimal(DS1307_Read(0x03));`

Následne je porovnávaný aktuálny čas s časom budíka (hodiny a minúty):

Podmienka: `if(bud_hod == hours && bud_min == minutes)`

Ak sa hodiny a minúty zhodujú, tak sa binárnou logikou a bitovými operáciami overí deň v týždni, keďže *master* zariadením bola odoslaná v jednom čísle hodnota všetkých aktivovaných dní (1 až 7), pre ktoré platí budík. V nasledujúcom kóde je zobrazené riešenie:

Binárne čítanie: `for(int i = 0; i < 7; i++)`

Kontrola dňa v týždni: `if(bud_dni & (1 << i) && (i + 1) == day_n)`

Následne *master* po aktivácii budíka odošle *slave* zariadeniam všetky potrebné premenné (povely zobrazené v tabuľke 4) po sériovej *USART_1* zbernici. Vo všetkých prípadoch sa posiela aj súčet všetkých odosielaných hodnôt premenných pomocou súčtu „*CHECKSUM*“.

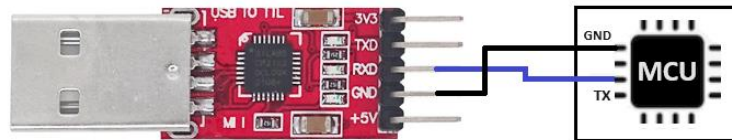
V nasledujúcej tabuľke sú zobrazené všetky povely, ktoré sa odosielajú *slave* zariadeniam.

Tab. 4 Kódy príkazov pre *slave* zariadenia.

1. bajt	2. bajt	3. bajt	4. bajt	5. bajt	6. bajt	7. bajt
0xB0	Teplota Interiér	Teplota Exteriér	vietor	Osvetlenie JUH	Osvetlenie ZAPAD	CHECKSUM
0xB1	Žalúzie ID	Pozícia	Natočenie	0	0	CHECKSUM
0xB3	Žalúzie ID	Pozícia	0	0	0	CHECKSUM
0xB5	Žalúzie ID	Minúty L	Minúty M	0	0	CHECKSUM
0xB6	Budík natočenie	Budík pozícia	Budík žalúzie ID	0	0	CHECKSUM
0xB7	Žalúzie ID	Umývanie	0	0	0	CHECKSUM
0xB8	Žalúzie ID	0	0	0	0	CHECKSUM

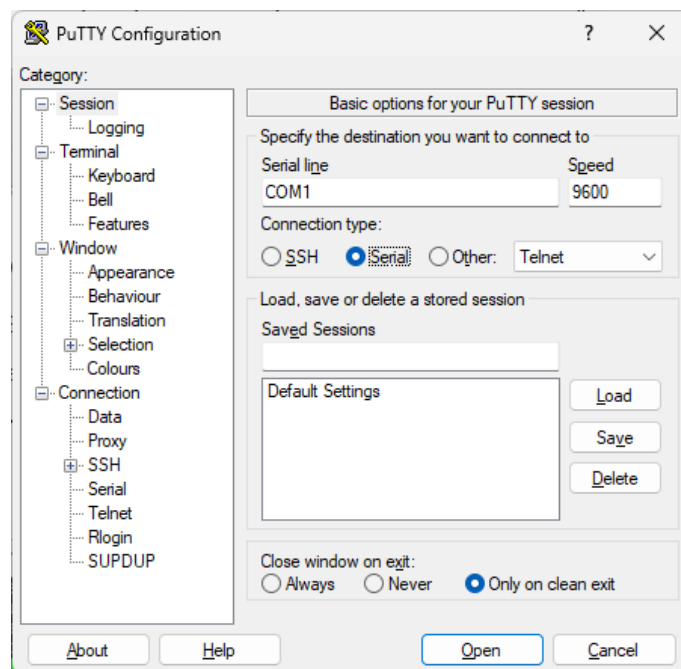
4.4. Overenie zaslaných údajov z *master* zariadenia do *slave* zariadení

Overenie odosielaných dát sa uskutočnilo pomocou prevodníka *USB-TTL*, ktorý po prepojení sériovej *USART_1* komunikácie (obr. 36) nahradí *slave* zariadenia, je akousi simuláciou pre testovanie funkčnosti sériovej linky a odosielaných dát *master* zariadením.



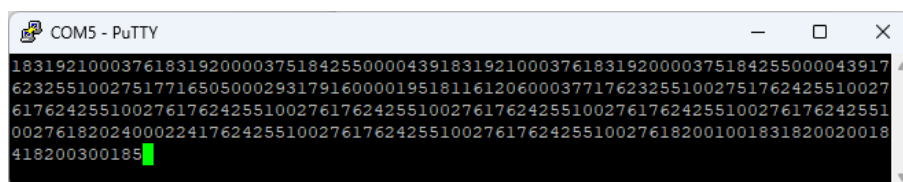
Obr. 36 Zapojenie *USART* prevodníka *USB/TTL* s *MCU*

Dáta prijímané *USB-TTL* prevodníkom sú zobrazované pomocou softvéru „*PUTTY*“, kde bola nastavená komunikácia, cez presný *COM* port.



Obr. 37 Softvér *PUTTY*

V terminálovom okne (obr. 38) sú zobrazené výsledky, ktoré *master* zariadenie odosiela ďalej pre *slave* zariadenia.



Obr. 38 Výstup odoslaných dát *master* zariadenia *slave* zariadeniam cez *TTL* (*PUTTY* terminál)

Záver

Cieľom tejto práce bolo navrhnúť a implementovať systém riadenia vonkajších žalúzií pomocou *Nextion* displeja a *master* zariadenia s mikrokontrolérom *ATMega324PB*. Počas realizácie projektu sme preskúmali rôzne aspekty automatizácie, užívateľských prostredí, displejov pre inteligentné domácnosti a identifikovali potrebu pre efektívne a užívateľsky prívetivé riešenia v tejto oblasti.

Na základe analýzy požiadaviek sme vyvinuli systém, ktorý umožňuje užívateľovi jednoduché a spoľahlivé riadenie vonkajších žalúzií prostredníctvom intuitívneho *Nextion* displeja. Implementácia tohto systému nám umožnila dosiahnuť hlavné ciele práce a po realizácii zlepšenie komfortu užívateľa a optimalizácie energetických nákladov.

Naša práca však taktiež odhalila niekoľko obmedzení a možných smerov pre ďalší výskum. Medzi tieto patria rozšírenie funkcionality systému o ďalšie senzory vonkajšej teploty, osvetlenia a sile vetra, ako aj optimalizácia spotreby energie *master* zariadenia pomocou *stand-by* režimu. S doplnením funkcionalít vonkajšej teploty, osvetlenia a sile vetra je potrebné dokončiť pre tieto senzory predpripravené funkcie v projekte. Projekt ďalej pokračuje nastavením a programovaním *slave* zariadení, ktoré budú riadiť už konkrétne žalúzie pomocou implementovaných príkazov.

Celkovo môžeme povedať, že práca priniesla cenné poznatky o riadení vonkajších žalúzií pomocou *Nextion* displeja a *master* zariadenia s mikrokontrolérom *ATMega324PB*. Veríme, že naše výsledky môžu byť užitočné pre ďalší vývoj v oblasti automatizácie domácností a inteligentných technológií.

Zoznam použitej literatúry

- [1]. prof. Ing. Daniela Perduková, PhD., Technická univerzita v Košiciach: Prednášky z predmetu tvorba používateľských rozhraní, [Online skriptá] [Dátum: 2024] Dostupné na webovej stránke: <<http://kem.fei.tuke.sk/sk/predmety/pouzivatelske-rozhrania>>
- [2]. HMI Human Machine Interface, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://inductiveautomation.com/resources/article/what-is-hmi>>
- [3]. GUI Graphical User Interface, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://coderama.com/slovník/gui>>
- [4]. Nextion, Description, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://nextion.tech>>
- [5]. Nextion, Basic Series Introduction, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://nextion.tech/basic-series-introduction>>
- [6]. Nextion, Discovery Series Introduction, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://nextion.tech/discovery-series-introduction>>
- [7]. Nextion, Enhanced Series Introduction, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://nextion.tech/enhanced-series-introduction>>
- [8]. Nextion, Intelligent Series Introduction, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://nextion.tech/intelligent-series-introduction>>
- [9]. Nextion, NX8048T050 datasheet, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://nextion.tech/datasheets/nx8048t050/#7>>
- [10]. Nextion, Editor Guide, [online] [Dátum: 2024] Dostupné na webovej stránke: <https://nextion.tech/editor_guide/#e3>
- [11]. Nextion HMI displeje, [online] [Dátum: 2024] Dostupné na webovej stránke: <<https://techfun.sk/blog/projekty/nextion-hmi-displeje-vytvorenie-grafiky-a-zopnutie-led-pomocou-dotykoveho-displeja>>
- [12]. Programujeme AVR v jazyku C, Bc. Ondrej Závodský, [online KNIHA] [Dátum: 2024] Dostupné na webovej stránke: <https://svetelektro.com/Download/avr_kniha.pdf>
- [13]. Datasheet ATmega324PB, [online PDF] [Dátum: 2024] Dostupné na webovej stránke: <<https://www.microchip.com/en-us/product/ATMEGA324PB>>
- [14]. Datasheet DS1307, [online PDF] [Dátum: 2024] Dostupné na webovej stránke: <<https://www.alldatasheet.com/datasheet-pdf/pdf/58481/DALLAS/DS1307.html>>
- [15]. Datasheet SHT31, [online PDF] [Dátum: 2024] Dostupné na webovej stránke: <<https://www.alldatasheet.com/view.jsp?Searchword=SHT31>>

Prílohy

- Príloha A: CD médium – diplomová práca v elektronickej podobe.
- Príloha B: *HMI* projekt naprogramovaný pomocou *Nextion* editora.
- Príloha C: Výstupný *HMI* súbor (.tft) z *Nextion* editora, pre displej.
- Príloha D: Zdrojové kódy projektu naprogramované v Microchip Studio pre MASTER.