

Dokumentácia k projektu AMAVET Smat HOUM (2.0)

Peter Riša, Dušan Šúňava

Contents

Ciele projektu	1
Povedzme si čo je Smart Home	2
Základná myšlienka projektu	2
Ako sme to urobili?	2
Funkcie domu	3
Web stránka	3
Ovládanie garáže	3
Automatické svietenie/tienenie	3
Zobrazovanie hodnôt	4
Ohrev/klimatizovanie miestnosti	4
Server	4
Programovanie	4
Rozvetvenie projektu	5
projekt je rozvetvený na 2 podprojekty	5
Projekt na platforme GitHub	5
Ako to funguje?	5
Aplikácia v reálnom svete	5
Budúcnosť projektu	6

Ciele projektu

- Jednoduchosť
- Dostupnosť

- Škálovateľnosť
 - Open-Source
-

Povedzme si čo je Smart Home

Smart Home je domácnosť, ktorá obsahuje zariadenia, ktoré ponúkajú majiteľovi zariadení určitú kontrolu nad funkciami domu a to buď z mobilného zariadenia, alebo počítača. Termín automatizácia pomenúva deje, ktoré sa vykonávajú bez prítomnosti človeka. Tým pádom sú robené automaticky na základe nejakého podnetu.

Základná myšlienka projektu

Základnou myšlienkou projektu bolo vytvoriť model inteligentnej domácnosti, ktorú by si ktokoľvek mohol vylepšiť podľa vlastných predstáv. K tomu sme doplnili zobrazovanie informácií v reálnom čase na web stránke a aj lokálne pomocou “Smart TV”. Podarilo sa nám to.

Ako sme to urobili?

Ako prvé bolo potrebné postaviť model domu. Na stavbu domu sme použili OSB dosky, plexisklo, farbu a skrutky, aj naše nervy. Dom sa skladá z úrovne 1 a úrovne 2. Medzi úrovňou 1 a 2 je priestor, ktorý je využitý na umiestnenie všetkej potrebnej elektroniky (počítač (server), ATX zdroj, Arduino, potrebné vodiče). Na druhej úrovni sa nachádzajú priečky, ktoré tvoria miestnosti. Dom obsahuje 3 miestnosti.

1. miestnosť - garáž. V garáži sa nachádza demonštrácia funkcie ovládania garáže s pomocou web stránky.
2. miestnosť - obývačka. V obývačke sa nachádza “Smart TV” zo zobrazovaním aktuálnych hodnôt zo senzorov a stave garáže v reálnom čase. Taktiež sa tu nachádza LED osvetlenie vzdialene ovládateľné.
3. miestnosť - spálňa. Slúži na demonštrovanie automatického tienenia (záclony), svietenia a mala slúžiť na demonštrovanie ohrevu/klimatizovania.

Model sme postavili relatívne rýchlo. No programovanie a zapojenie elektroniky trvalo nejaký čas. Aby nám fungovalo riadenie zariadení a monitorovanie senzorov sme potrebovali vybrať zariadenie s množstvom digitálnych aj analógových vstupov/výstupov. Na túto úlohu sme zvolili Arduino Mega 2560, kvôli

neprekonateľnému pomeru cena/funkcie. Ako server, ktorý robí uzol medzi užívateľom a zariadeniami sme zvolili Raspberry Pi 3B+ (neskôr plnohodnotný X86 server). Potrebovali sme len malé zariadenie na ktorom pobeží NodeJS server s modulom npm. (neslôr sa ukázalo ako nedostačujúce) Vyvinuli sme taktiež jednoduchú web stránku, ktorou je možné ovládať všetky funkcie v modeli. Ako napájanie slúži 250W mATX zdroj.

Funkcie domu

Web stránka

Stránka je urobená tak, aby bola prehľadná. Jej kód je napísaný v HTML(HBS), CSS a JavaScripte. Backend (serverová časť) aj frontend (užívateľské prostredie) je urobené od základu na mieru. Má jednoduché a intuitívne prostredie. Zámerne má tmavšie prostredie, aby počas noci, alebo skorých ranných hodín nerušila bielym pozadím. Farba tlačidiel sa mení v závislosti od stavu prislúchajúcich koncových polôh, resp. stavu zariadenia. Zmena farieb funguje ako signalizácia stavu zariadenia - nie je nutné ísť k zariadeniu a kontrolovať, či sa príkaz naozaj vykonal. Odozva od stlačenia tlačidla na web stránke a vykonaním akcie je 300ms. Odozvu je možné ešte znížiť, ak by to bolo nutné.

Ovládanie garáže

Garáž sa ovláda pomocou prehľadnej webovej stránky. Mechanika je urobená zo závitovej tyče, DC motora, ložiska, tiahla a brány. Brána je zostrojená ako sendvič. Uprostred je plastová doštička, na ktorú je prilepený drôtik, ktorý dodáva pevnosť. Flexibilitu umožňuje silná sivá lepiaca páska, alebo “duct tape”. Motor roztáča závitovú tyč, na ktorej je tiahlo, ktoré pohybuje s bránou. Tým ju zatvára a otvára. Aby sme docielili zatváranie aj otváranie, bolo nutné meniť polaritu motora. Menenie polaritu funguje pomocou PWM drivera L298N ovládaného Arduinom. Aby sazariadenie nepoškodilo boli pridané mechanické koncové spínače. Spínače pracujú takmer okamžite.

Automatické svietenie/tienenie

Svetlá majú tri možnosti fungovania: zapnúť / vypnúť / automatika. Ovládanie prebieha pomocou tlačidiel na web stránke. Spínanie funguje pomocou relé, ovládané je Arduinom. Ako svetlá boli použité biele 12V LED pásy. Automatický mód funguje pomocou senzoru okolitého jasu. Údaje zo senzoru vyhodnocuje Arduino. Ak dostane pokyn od serveru na zmenu stavu na automatiku, tak Arduino začne aktívne porovnávať údaje zo senzorov a prednastavenou hodnotou pre zapnutie svetiel. Momentálne je nastavená hodnota 20% pre svetlá a 40% pre závesy, ak klesne úroveň okolitého jasu pod túto úroveň, tak svetlá sa zapnú a závesy sa otvoria. Naopak, ak sa okolitý jas dostane nad úroveň 20%, resp. 40%

tak sa svetlá vypnú a závesy sa zatvoria. Závesy fungujú iba v automatickom režime.

Zobrazovanie hodnôt

Zobrazovanie hodnôt sa odohráva na 3.2" displeji, ktorý je napojený na Arduino. Displej zobrazuje stav svetiel, okolitý jas, otvorenie/zatvorenie garáže, nastavenú teplotu a aktuálnu teplotu okolia. Prekresľovanie hodnôt sa deje iba pri zmene stavu. Tým je ušetrený výpočtový výkon na iné procesy.

Ohrev/klimatizovanie miestnosti

V kóde projektu je zabudovaná logika pre nastavenie ohrevu/klimatizovania miestnosti. Táto možnosť je momentálne len teoretická, z dôvodu poruchy jedného z dvoch TEC (peltierových) článkov. S jedným článkom nebolo možné dosiahnuť dostatočne viditeľný efekt na teplotu miestnosti. Ohrev/klimatizovanie malo prebiehať pomocou TEC (peltierových) článkov, na ktorých by sa nachádzali chladiče zo starých procesorov. Za pomoci prúdenia vzduchu z ventilátorov cez otvory v podlahe by bol do miestnosti vháňaný vzduch. Táto možnosť je momentálne vo vývoji.

Server

Pod pojmom server je myslený počítač na ktorom beží web stránka. Jedná sa o počítač, na ktorom beží GNU/linux-ová distribúcia ArchLinux. Server obsahuje skript, ktorý automaticky po nabenutí počítača zapne server, aby bolo možné sa naň pripojiť. Funguje to vďaka nastaveniu funkcie v systemd(sysctl - automatický proces, ktorý je možné povoliť po nabenutí systému GNU/linux). Počítač je nakonfigurovaný tak, aby sa nedal len tak jednoducho zneužiť. Je v ňom aktivovaný účinný firewall. Taktiež server má obmedzený vstup na niektoré stránky. Je to vyriešené pomocou zmeny konfigurácie súboru hosts (nachádza sa v /etc/hosts), resp. pridaním zoznamu stránok, ktoré majú nastavené hostovské meno a ip na 0.0.0.0 a 127.0.0.1. Vďaka tomuto sa počítač snaží po zadaní určitej adresy, ktorá sa nachádza v zozname pripojiť sám na seba. To vyústi do chyby v pripojení. Bola upravená konfigurácia SELinux-u. Jediné teoretické riziko je SSH. SSH používa iný ako základný port (22), z dôvodu zabezpečenia.

Programovanie

Programovanie prebiehalo v prostredí programu VScode. Neskôr počas vývoja sme prešli z Raspberry Pi 3B+ (arm64) na intel X86 platformu. Bolo to z dôvodu potreby vyššieho výkonu. Začali sme vyvíjať projekt na diaľku. Na zariadení Raspberry by kompilácia projektu a následné nahratie do Arduina trvala nesmierne dlho. V kostre modelu je preto zabudovaný malý pasívne chladený

počítač spolu s mechanickým diskom a zdrojom. Počítač má dostatočný výkon na daný účel.

Rozvetvenie projektu

projekt je rozvetvený na 2 podprojekty

1. “dom_displej” - časť projektu určená pre platformu Arduino. Obsahuje kompletný kód v jazyku C (C++), nevyhnutné knižnice a schémy. Projekt je prehľadný a jednoduchý. Obsahuje nápovedné komentáre ku každej časti kódu pre pochopenie. Kód je priebežne aktualizovaný, zjednodušený a zefektívnený. Jednotlivé časti kódu sú rozdelené do knižníc, ktoré sú volané iba ak sú potrebné. To umožňuje kódu pracovať efektívnejšie a rýchlejšie. Kód je otvorený každému záujemcovi.
2. “pi_server” - časť, ktorá má na starosti samotný server, ktorý beží na serveri. Obsahuje dokumentáciu, príručku ako ho spustiť, samotný kód stránky aj servera. Odporúčame ho spúšťať na GNU/linux-e. Nebol testovaný na platforme windows, alebo OS X. K správnej funkcii potrebuje program NPM spolu s modulmi, ktoré sú uvedené v projekte na platforme GitHub.

Projekt na platforme GitHub

<https://github.com/peterrisa/AMAVET>

- tu sa nachádzajú všetky potrebné dokumenty a odkazy

Ako to funguje?

Arduino je prepojené zo serverom pomocou univerzálnej sériovej linky. Zariadenia si vymieňajú každých 50 milisekúnd “telegramy”. Funkcia “telegram” je urobená s tým, že v budúcnosti možno bude potrebné prenášať viac informácií naraz. Webová stránka zobrazuje príchodzie informácie užívateľovi. Stránka taktiež vyhodnocuje logiku príkazov, aby sa zbytočne neposielali príkazy explicitne.

Aplikácia v reálnom svete

- Projekt je možné použiť aj v reálnom svete. Na použitie v reálnom svete je momentálne pripravená časť s automatickým svietením a tienením. Časť so zatváraním / otváraním garáže je potrebné prispôsobiť na konkrétnu garáž. Základná logika sa však meniť nemusí.

- Jedná o riešenie DIY (urob si sám), čo spotrebitelia nemusia oceniť, ale nadšenci áno.
 - Naše riešenie nezbiera žiadne informácie o užívateľoch, čo môže byť pre niekoho plusom. Ďalším faktom je, že si nadšeneci môžu kombináciou viacerých projektov vrátane nášho privyrobiť.
-

Budúcnosť projektu

- modularita a podpora pre rôzne druhy zariadení a senzorov
- jednoduché programovanie zariadení
- pridanie obsiahlych knižníc s návodmi a vylepšené komentáre v kóde
- uľahčenie programovania bežným ľuďom