



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ



ФАКУЛТЕТ ПО ЕЛЕКТРОННА ТЕХНИКА И ТЕХНОЛОГИИ

ДИПЛОМНА РАБОТА

**ТЕМА: МОБИЛНА СИСТЕМА ЗА
 НАБЛЮДЕНИЕ**

ДИПЛОМАНТ: Иван Петров Учърджиев

РЪКОВОДИТЕЛ: доц. д-р инж. Емил Димитров

София, 2016

ДЕКЛАРАЦИЯ

Долуподписаният: Иван Петров Учеджиев , декларирам, че съм разработил самостоятелно представената работа. Не съм използвал източници или ресурси, освен посочените. Ясно съм обозначил използваните източници, които са цитирани буквално или по съдържание.

Декларирам и че работата не е представяна в рамките на друга дипломна защита.

Дата:

Подпис

СЪДЪРЖАНИЕ

ДЕКЛАРАЦИЯ.....	3
УВОД.....	6
Глава 1. СРЕДСТВА ЗА ОСЪЩЕСТВЯВАНЕ НА АВТОМАТИЗИРАН КОНТРОЛ В ЕЛЕКТРОНИКАТА.....	7
1.1 Общи сведения.....	7
1.2 Система с цифрово-програмно управление.....	8
1.2.1 Основни задачи на системите с цифрово-програмно управление.	10
1.3 Микроконтролери.....	12
1.3.1 Принцип на работа на микроконтролерите.....	14
Глава 2. СЪВРЕМЕННИ ПРОЦЕСОРНИ ЯДРА ЗА МИКРОКОНТРОЛЕРИ.....	17
2.1 Общи сведения.....	17
2.1 Подробен преглед на „Cortex M” серията.....	17
Глава 3. РАЗРАБОТВАНЕ НА МОБИЛНА СИСТЕМА ЗА НАБЛЮДЕНИЕ.....	25
3.1 Блокова схема и обяснителна записка.....	25
3.2 Проектиране на модула за управление и модула за комуникация. .	28
3.2.1 Принципна схема на мобилната система за наблюдение.....	28
3.2.2 Разчет и съображения за избиране на елементите и техните стойности.....	28
3.4 Алгоритъм на работа на мобилната система за наблюдение.....	41
3.4.1 Алгоритъм на работа на главното управляващо устройство.....	41
3.4.1.1 Безжична комуникация.....	41
3.4.1.2 Алгоритъм на работа на основния процес на програмата на главното управляващо устройство.....	43
3.4.1.3 Алгоритъм на работа на комуникационния процес на програмата на главното управляващо устройство.....	47

3.4.2 Комуникация между главно управляващо устройство и микроконтролер.....	52
3.4.3 Алгоритъм на работа на микроконтролера.....	54
3.4.4 Алгоритъм на работа на потребителското устройство.....	61
3.4.4.1 Алгоритъм на работа на подпрограмата за свързване.....	61
3.4.4.2 Алгоритъм на работа на подпрограмата за управление.....	67
3.4.4.3 Алгоритъм на работа на подпрограмата за измерване.....	71
Глава 4. УКАЗАНИЯ ЗА НАЧИН НА РАБОТА С МОБИЛНАТА СИСТЕМА ЗА НАБЛЮДЕНИЕ.....	76
ЗАКЛЮЧЕНИЕ И НАСОКИ НА РАЗВИТИЕ НА СИСТЕМАТА.....	82
ЛИТЕРАТУРА.....	83

УВОД

Мобилните системи и роботите се използват широко в днешно време. Те допринасят за подобряване на битовата среда на човек, както и промишлената и производствена дейност на големи компании. Такива системи масово се проектират и използват за изследване на опасни среди за човек, както и за извършването на дейности, вредни или непосилни за хората. Развитието на телекомуникациите и телекомуникационните системи позволява осъществяването на предаване в реално време на видеоизображения, сензорни измервания и управление на мобилни системи по жичен или безжичен път.

Целта на тази дипломна разработка е проектирането на мобилна система за наблюдение в реално време с възможност за управление от мобилен телефон по безжичен интерфейс. Подобен тип система може да бъде имплементирана в различни проекти (примерно като част от проект за домашна автоматизация) и използвана за различни цели (например за изследване и измерване параметрите на среда, опасна за човек).

Глава 1. СРЕДСТВА ЗА ОСЪЩЕСТВЯВАНЕ НА АВТОМАТИЗИРАН КОНТРОЛ В ЕЛЕКТРОНИКАТА

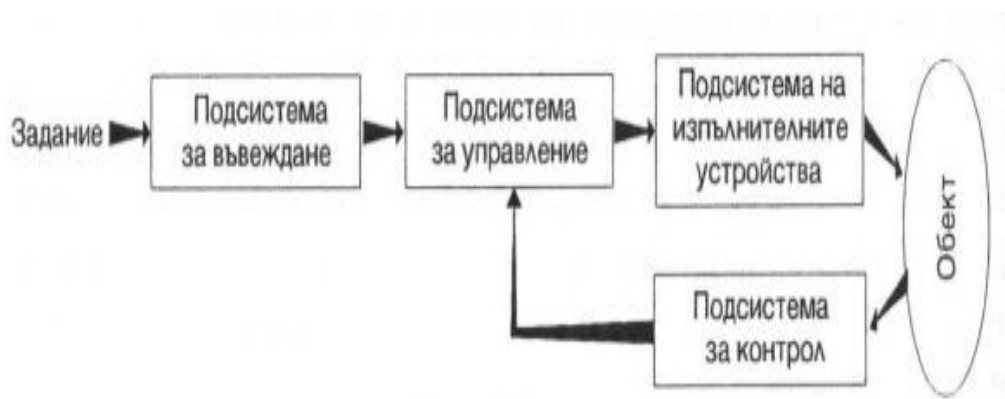
1.1 Общи сведения

В днешно време автоматизираният контрол се използва широко с различно предназначение. Целта на автоматизирания контрол е повишаване качеството на извършвания технологичен процес, по-голямо бързодействие, по-висока производителност и др. Автоматизацията се извършва от системи с цифрово-програмно управление. Такава система се нарича технологично съоръжение, което изпълнява своите функции автоматично под въздействието на управляваща програма, представляваща поредица от символи, записани на програмоносител. Цифрово-програмно управление се използва основно при:

- Автоматизация на управлението на превозни средства;
- Автоматизация на изготвянето на конструктивна и технологична документация;
- Автоматизация на технологичните операции, използвани при технологичната подготовка за производство и при самото производство – изготвяне на фотошаблони, металорязане, разтапяне, щанцоване, поене, лазерни технологии и др.;
- Автоматизация на контролно-измервателни операции, определяне на размери на детайли, измерване на електрически параметри, измерване на разстояние, скорост, налягане и др.;
- Автоматизация на диагностични системи – за откриване на дефекти и грешки в електронни изделия.
- Управление на промишлени и битови роботи – за контрол на мотори, събиране на данни от сензори, анализ на информация, вземане на решения за действия и др.

1.2 Система с цифрово-програмно управление

Системите с цифрово-програмно управление са автоматично работещи устройства. За да се постигнат определени количествени и качествени резултати чрез управление на технологичната част, е необходимо в системата за управление да има специализирани технически средства, групирани в четири основни подсистеми. На фиг.1.1 е показана блокова схема на подсистемите, използвани в системата за управление.



Фиг.1.1 Блокова схема на подсистемите използвани в системата за управление, (по Овчаров, С.: 2008)

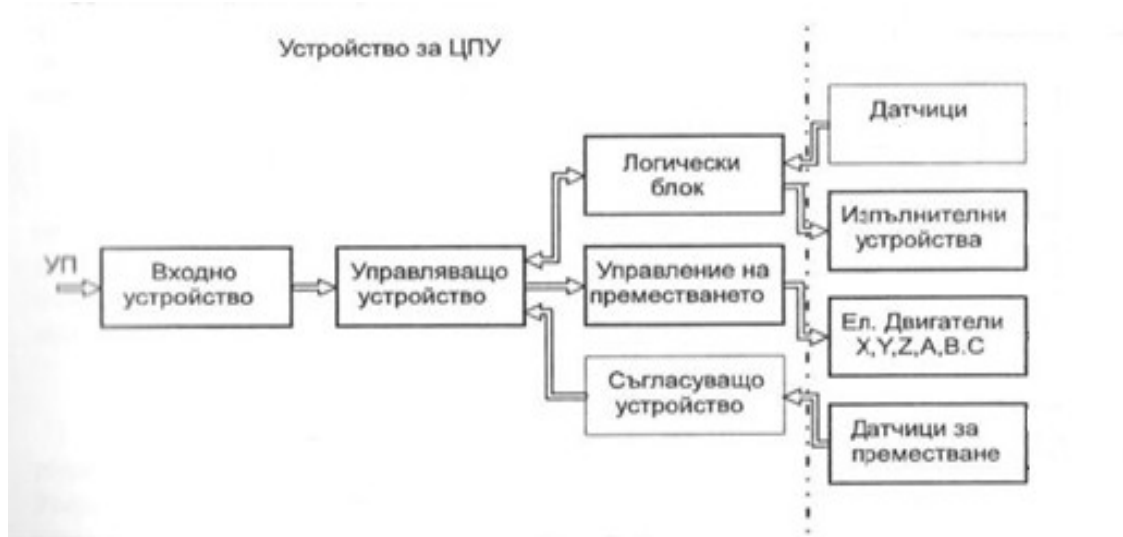
1. Подсистема за въвеждане – използва се за въвеждане на задание, съдържащо информация за величините и техните стойности, характеризиращи технологичната част (обекта на управление);

2. Подсистема за контрол – изработва информация за състоянието на обекта, изразяваща се в стойности на величините, характеризиращи обекта;

3. Подсистема за управление – в тази подсистема се обработва информацията, постъпваща от подсистемата за въвеждане и подсистемата за контрол и се изработва информация за управление (управляващи сигнали) на подсистема на изпълнителните устройства;

4. Подсистема на изпълнителните устройства – преобразува информацията за управление в електрически сигнали за задействане на изпълнителните устройства, с които се въздейства върху обекта, [3].

На фиг.1.2 е представена блокова схема на система с цифрово-програмно управление .



Фиг.1.2 Блокова схема на устройство с цифрово-програмно управление,
(по Овчаров, С. :2008)

Устройството за ЦПУ (цифрово-програмно управление) е част от системата с цифрово-програмно управление и представлява конструктивно завършена единица, която управлява системата. Това устройство се реализира на базата на микроконтролери и служи за изработване на управляващи сигнали по зададена програма.

Управляващата програма съдържа информация за величините и техните стойности, характеризиращи технологичната част (обекта на управление) и инструкции за управление на системата. Управляващата програма се въвежда в паметта на системата с помощта на входното устройство (ВУ). Управляващото устройство (УУ) чете информацията в паметта ред по ред. То изработва управляващи сигнали на базата на информацията, прочетена от паметта, които праща към логическия блок

(ЛБ). Логическия блок изработва електрически сигнали за управление на изпълнителните устройства. Управляващото устройство също така изработва сигнали за управление на движение, които подава към блока за управление на двигателите, който изработва необходими сигнали за завъртане на двигателите. Следенето на движението на двигателите се осъществява от сензори за преместване, [3].

1.2.1 Основни задачи на системите с цифрово-програмно управление.

Основните задачи, които имат системите с цифрово-програмно управление, са следните:

1. Въвеждане и съхранение на системно-програмно осигуряване – функционалните възможности на дадена система с цифрово-програмно осигуряване зависят от вграденото в нея системно-програмно осигуряване (съвкупност от всички инструкции и данни, необходими за работа на всяка електронно-изчислителна машина). Въвеждането на това програмно осигуряване обикновено се извършва еднократно при производството на системата. Повторно въвеждане се налага след тежки аварии, когато е разрушена използваната за съхранение енергонезависима памет. При някои системи се използва енергозависима памет от тип RAM с батерийно захранване. При този тип системи има възможност при необходимост да се извърши смяна на системно-програмното осигуряване с цел усъвършенстване на функционалните възможности на системата.

2. Въвеждане и съхраняване на управляваща програма – често начина на изпълнение на функциите на системата с цифрово-програмно управление се налага да се обновява или сменя. Поради тази причина в съвременните системи управляващата програма се въвежда и съхранява в памет от тип EEPROM (електронно-изтриваема програмируема памет).

При стартиране на системата, програмата се въвежда в RAM паметта на системата за по-бързо и по-лесно изпълнение.

3. Реализация на цикли – това са участъци от управляващата програма, които се изпълняват често и се оформят във вид на подпрограми за извършване на определена последователност от инструкции. Специализираните системи с цифрово-програмно управление имат т.нар. „фиксиращи цикли“, които не са въведени в управляващата програма, а в системно-програмното осигуряване с цел намаляване на обема на управляващата програма.

4. Интерпретация на кадъра – управляващата програма се състои от изречения, чиито физически запис се нарича кадър. Преди да се пристъпи към обработване на поредния кадър, е необходимо върху информацията, съдържаща се в него, да се извършат определени действия (преобразуване в двоичен код, сортиране по регистри и др.). Тези действия трябва да се завършат преди завършването на изпълнението на предишния кадър.

5. Интерполация – използва се когато системата управлява движение. В резултат от интерполацията се получават координати на междинните точки на дадена траектория, зададена по начална, крайна точка и вид на траекторията.

6. Управление на движения – използва се когато системата управлява движение. Тази задача се свежда до извършване на преместване в точките, чиито координати са изчислени при изпълнението на интерполацията. Управлението на движение се реализира чрез цифрови следящи системи или т.нар. „моторни драйвери“, чиито входове се управляват с цифрови сигнали, а на изходите им се получава сигнал за управление на изпълнителен двигател. За следене на движението се използват сензори, които подават информация за положението на обекта на движение.

7. Логическо управление – този вид управление е свързано с изпълнение на команди от типа „включи-изключи“. Тяхното изпълнение е

свързано с проверка на условията, свързани със състоянието на обекта или с отчитането на времеви съотношения, свързани с обработването на закъснения. Задачата по изпълнението на дадена команда се свежда до проверка на условие, характеризиращо начина на реакция при определено събитие. Ако резултатът от проверката е „истина“, се реагира по определен начин и се изработват сигнали за изпълнение на командата; ако резултатът е „лъжа“, се връща сигнал към управляващото устройство, което реагира по определен начин в зависимост от характера на неизправността.

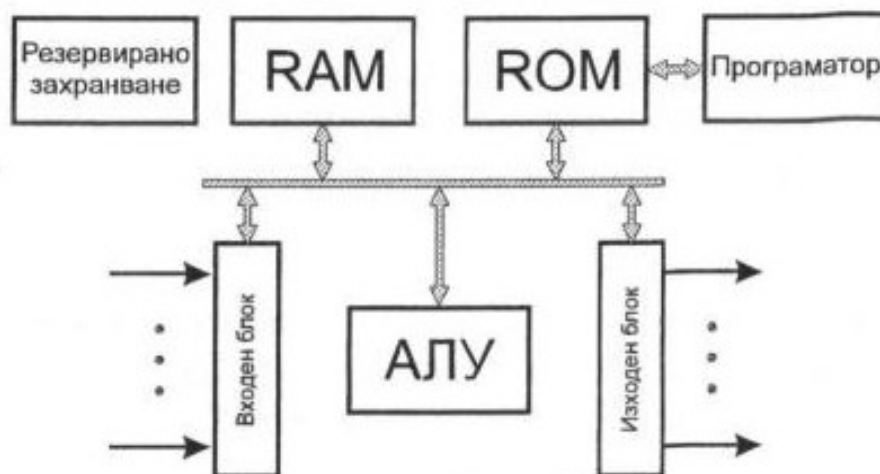
8. Адаптивно управление – с цел повишаване на качеството на управление и точността се налага да се изменят някои параметри на управляващата програма (пр.: стъпка, скорост на движение и др.). Адаптивното управление се изпълнява чрез въвеждане на допълнителни сензори или алгоритми в управляващата програма, които да следят качеството на управление. Информацията, използвана за адаптивно управление може да бъде различна в зависимост от предназначението на системата, но основни примери са: атмосферни изменения на околната среда, изменение на електрическите параметри и др. Информацията се подава към управляващото устройство, което изменя съответните параметри на управляващата програма, за да се приспособи системата към новите условия.

9. Комуникационен модул – често в днешно време се налага съвместна работа на повече от една система с цифрово-програмно управление. Поради тази причина тези системи трябва да съдържат стандартни комуникационни интерфейси за реализиране на комуникацията между отделните системи, [3].

1.3 Микроконтролери

Съвременните системи за управление, използвани за автоматизиране, се изграждат на базата на универсални микропроцесорни устройства или

микроконтролери. Тези контролери представляват електронни устройства, управлявани програмно. Управлението на дадена система от микроконтролер се осъществява, като се обработва информация от цифрови и аналогови входове и се изработва информация, подавана на цифрови и аналогови изходи. Информацията от входовете служи за въвеждане на стойности на определени параметри или характеристики, които са цел на управлението, а информацията от изходите се използва за задействане на информационни и изпълнителни устройства, служещи за въздействие върху обекта. Обработката на информация се управлява от програма, записана в паметта на контролера. На фиг.1.3 е представена основна блокова схема на микроконтролер, [1; 2; 3].



Фиг. 1.3 Блок схема на микроконтролер (по Овчаров, С.:2008)

Основните блокове на микроконтролерите са:

- Захранване – трябва да осигури захранването на всички модули в микроконтролера. Този блок също така трябва да осигури защита от външни смущения и резервиране на захранването на оперативната памет на контролера;
- Процесорен модул – изграден е на базата на стандартна микропроцесорна система, като в модула са включени в повечето случаи само процесора и паметта;

- Входни модули – те могат да бъдат за обработване на логически сигнали и на аналогови сигнали. Препоръчително е във входните вериги да има галванично развързване между източника на входния сигнал и останалите модули. Аналоговите сигнали преди подаване към входните модули трябва да се нормират (да се съгласуват със стандартните напрежителни и токови нива на входните модули). Аналоговите входни модули могат да бъдат за обработване на еднополярно или двуполярно напрежение;

- Изходни модули – биват за логически и аналогови сигнали. Препоръчително е да се осигури галванично развързване между микроконтролера и товара;

- Специализирани модули – предназначени са да обработват информация, постъпваща от устройства с нестандартни изходи или да изработват информация за управление на изпълнителни механизми с нестандартни сигнали. Обикновено това са аналогови входове и изходи, които са свързани към възли за мащабиране, филтрация, линеаризиране, дискретизация и др. Към тези модули се отнасят и устройства за обработка на сигнали от специализирани източници, като например фоторастерни преобразуватели, използвани за отчитане на преместване на електромотори;

- Комуникационни модули – осигуряват възможност на контролера да обменя информация с други системи или устройства, [1; 2; 3; 4].

1.3.1 Принцип на работа на микроконтролерите

Микроконтролерите изпълняват своите функции, като се направляват от информация, записана на паметта им. Паметта им се дели на програмна памет (енергонезависима ROM, EPROM, EEPROM) и оперативна памет (енергозависима RAM). В програмната памет се съхранява управляващата

програма. Управляващата програма обикновено се дели на три основни модула:

- Модул за начално установяване – изпълнява се еднократно след постъпване на определено събитие, като например команда за стартиране на работата на микроконтролера;

- Основен модул – тук се реализира основната част от алгоритъма за управление на обекта. Обработват се логически данните за обекта и тези, съдържащи се в управляващата програма. Модулът се изпълнява циклично по време на т.нар. „оперативен цикъл“;

- Модул „прекъсващи програми“ – прекъсващите програми осигуряват бърза реакция на контролера при определени обстоятелства. Този тип програми прекъсват изпълнението на основния модул, като текущото му състояние се прехвърля в т.нар. „стек“ или „стекова памет“. След записа на текущото състояние на основния модул в стековата памет се изпълнява функцията на прекъсващата програма, съответстваща на даденото обстоятелство. При приключване на изпълнението на прекъсващата програма контролера се връща към изпълнение на основния модул, като извлича информация до къде е стигнало изпълнението от стековата памет.

Режимът „реално време“ на контролерите се характеризира с осигуряване времето за реакция на дадено събитие да бъде по-малко от едно предварително зададено време. Тази функция се постига чрез използване на механизмите за прекъсване, вградени в микропроцесорните системи и микроконтролерите и т.нар. „режим на времеделение“ (за всяка задача се осигурява периодично определен времеви интервал). Основните функции на системите за реално време са:

- Управление на задачите – чрез тази функция се разпределя времето между отделните задачи и е свързана с блокиране (прекъсване) и деблокиране (възстановяване) на изпълнението на задачите в съответствие с настъпването на определени събития;

- Управление въз основа на приоритетите – задачите, изпълнявани от контролера, се подреждат по приоритет, който зависи от допустимото време за реакция на предизвикващото ги събитие. В повечето случаи приоритета е фиксиран, но в някои случаи може да се променя по време на изпълнението на програмата;
- Обмен на информация между задачите – използват се определени участъци от паметта, в които се поставя информация и данни, общи за даден набор от задачи;
- Разпределение на ресурсите – тази функция е свързана с правилата на ползване на времето на процесора, обема на оперативната памет, времето на комуникационните модули и др., [1; 2; 3].

Глава 2. СЪВРЕМЕННИ ПРОЦЕСОРНИ ЯДРА ЗА МИКРОКОНТРОЛЕРИ

2.1 Общи сведения

Благодарение на развитието на микроелектрониката, производителността на микроконтролерите значително нараства през последните години. Масово големите производители на микроконтролери, като „STMicroelectronics”, „Texas Instruments”, „NXP”, „Renesas” и други, преминаха от собствени 8 и 16 битови архитектури на „ARM” базирана 32 битова архитектура. „ARM” предлага фамилия от 32 битови процесорни ядра наречена „Cortex M”. Масовото използване на тази фамилия води до унифициране на микроконтролерните ядра, което позволява лесно преминаване от един контролер на друг, както и лесно имплементиране на операционни системи за реално време. Фамилията микропроцесорни ядра „Cortex M” е пригодена за лесно програмиране на програмния език „C”, което позволява по-бързо и лесно писане на програми за тези ядра.

2.1 Подробен преглед на „Cortex M” серията

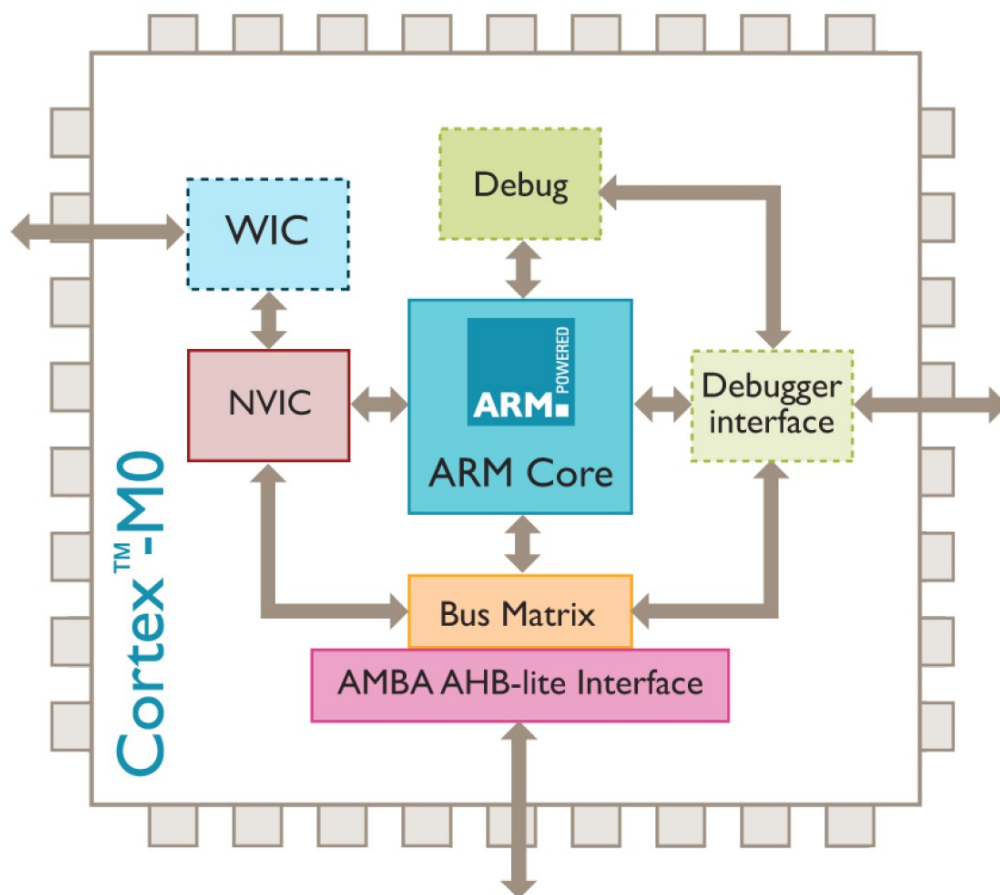
Микропроцесорните ядра от „Cortex M” фамилията, представляват специално проектирани ядра за внедряване в микроконтролери. До момента представителите на тази серия са следните:

1. „Cortex M0” - това процесорно ядро е най – малкото ядро проектирано от „ARM”. То е пригодено за приложения, където консумацията е критична. Основните параметри на „Cortex M0” ядрото са следните:

- Консумация – 85 μ W/MHz;
- Фон Нойменова архитектура;
- Производителност – 0.9 DMIPS/MHz;

- Интегриран контролер на прекъсвания – 1,8,16,24 или 32 вектора за прекъсвания;
- Контролер за събуждане;
- Системен брояч;
- Дебъгване – 4 или 2 точки на прекъсване, 2 или 1 точки за наблюдение, JTAG или SWD интерфейс.

На фиг. 2.1 е представена блокова схема на процесорното ядро[5].



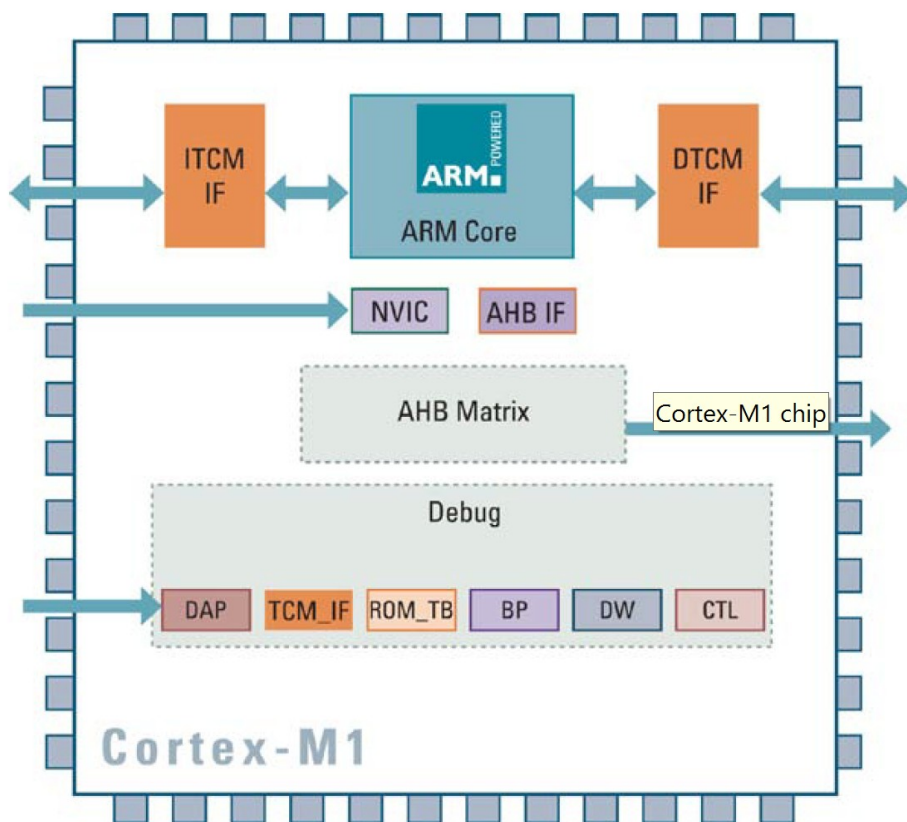
Фиг. 2.1 Блокова схема на микропроцесорно ядро “Cortex M0”

2. “Cortex M1” - този процесор се проектиран за вграждане в чипове с програмируема логика „FPGA”. Характеризира се с следните основни параметри:

- Производителност – 0.8 DMIPS/MHz;
- Максимална тактова честота – 200 MHz;

- Архитектура на процесора – RISC;
- Интегриран контролер на прекъсвания – 1,8,16,24 или 32 вектора за прекъсвания;
- Програмируем приоритет на прекъсванията – от 1 до 4;
- Дебъгване – 4 или 2 точки на прекъсване, 2 или 1 точки за наблюдение, JTAG или SWD интерфейс[6].

На фиг. 2.2 е представена блокова схема на процесорното ядро.



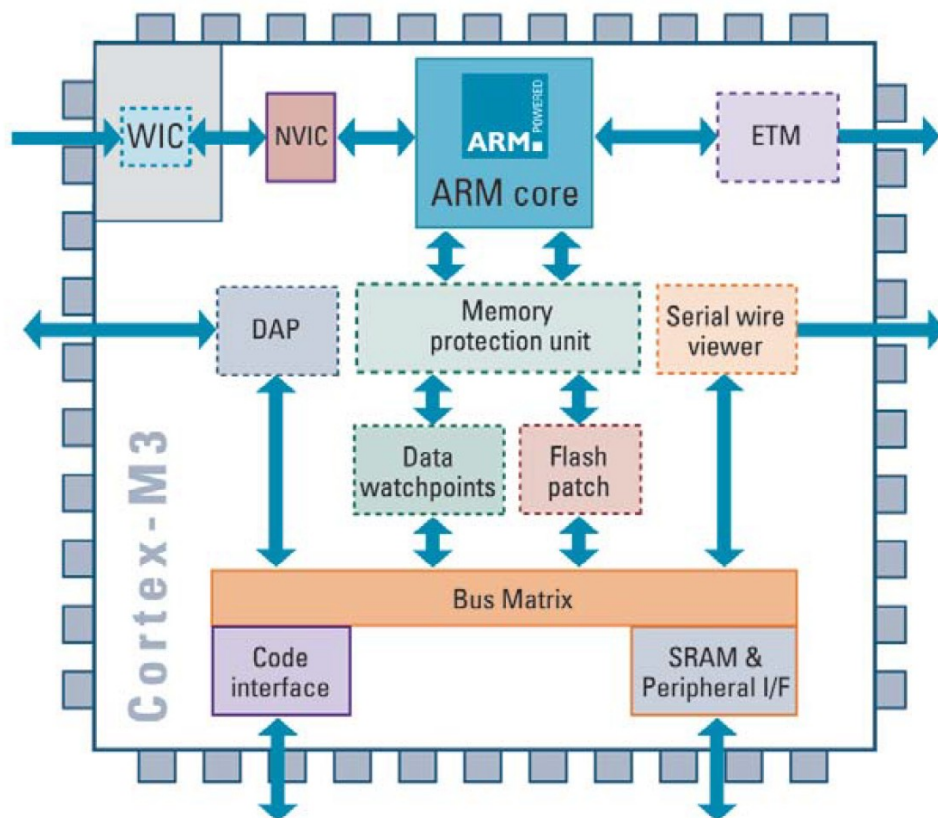
Фиг. 2.2 Блокова схема на микропроцесорно ядро „Cortex M1”

3. “Cortex M3” - това микропроцесорно ядро е проектирано за внедряване в системи, които изискват по-висока изчислителна производителност. Характеризира се със следните основни параметри:

- Производителност – 1.25 DMIPS/MHz;
- Харвардска архитектура;

- Устройство за защита на паметта „MPU” ;
- Интегриран контролер на прекъсвания – до 240 вектора за прекъсвания;
- Програмируем приоритет на прекъсванията – от 8 до 256;
- Контролер за събуждане;
- Системен Брояч;
- Дебъгване – до 8 точки на прекъсване, до 4 точки за наблюдение, JTAG или SWD интерфейс.[7]

На фиг. 2.3 е представена блокова схема на процесорното ядро.



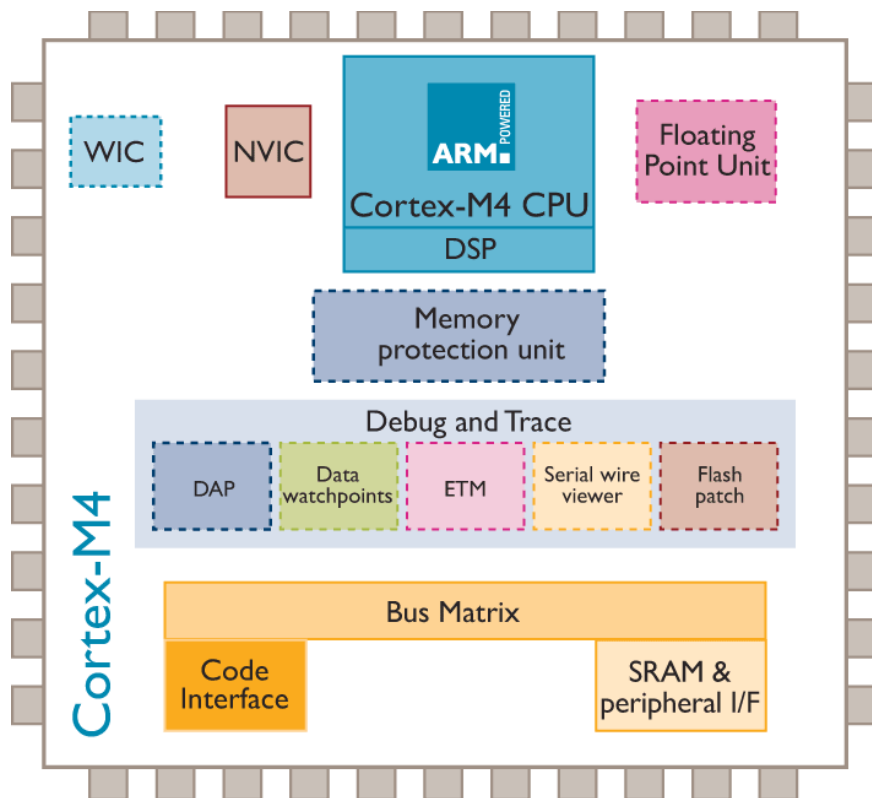
Фиг. 2.3 Блокова схема на микропроцесорно ядро „Cortex M3”

4. “Cortex M4” - това микропроцесорно ядро, представлява „Cortex M3” ядро плюс поддръжка на „DSP” инструкции и възможност за смятане с плаваща запетая. Характеризира се със следните основни параметри:

- Производителност – 1.25 DMIPS/MHz;

- Устройство за защита на паметта „MPU” ;
- Възможност за работа с 64 битови числа;
- Интегриран контролер на прекъсвания – до 240 вектора за прекъсвания;
- Програмируем приоритет на прекъсванията – от 8 до 256;
- Контролер за събуждане;
- Системен Брояч;
- Дебъгване – до 8 точки на прекъсване, до 4 точки за наблюдение, JTAG или SWD интерфейс.[8]

На фиг. 2.4 е представена блокова схема на процесорното ядро.

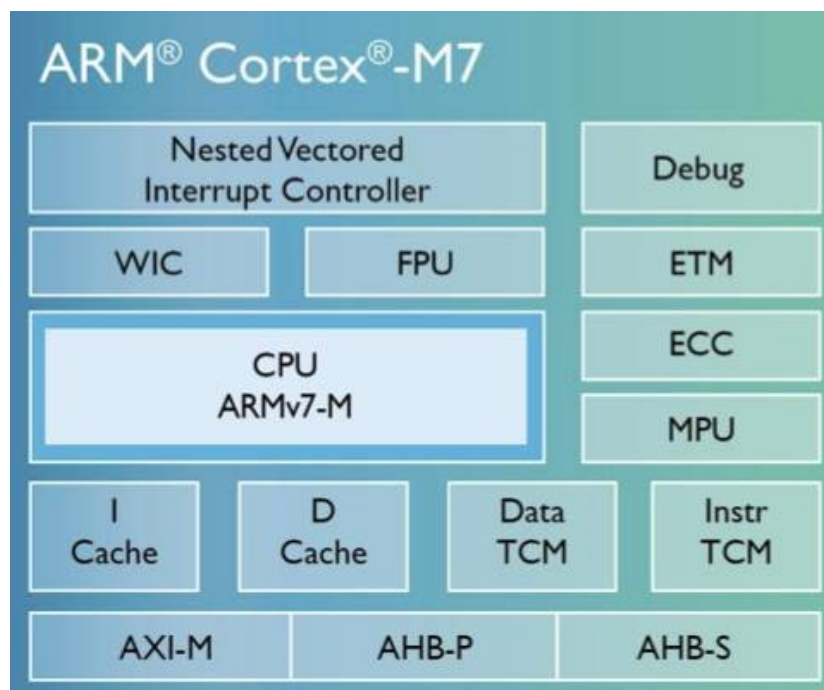


Фиг. 2.4 Блокова схема на микропроцесорно ядро „Cortex M3”

5. “Cortex M7” - това микропроцесорно ядро е най-новият представител на фамилията „Cortex M” и най – високо производителното ядро от тази фамилия. Характеризира се със следните основни характеристики:

- Производителност – 2.14 DMIPS/MHz;
- Архитектура - RISC;
- Устройство за защита на паметта „MPU”;
- Възможност за работа с 64 битови числа;
- Интегриран контролер на прекъсвания – до 240 вектора за прекъсвания;
- Програмируем приоритет на прекъсванията – от 8 до 256;
- Контролер за събуждане;
- Системен брояч;
- Дебъгване – до 8 точки на прекъсване, до 4 точки за наблюдение, JTAG или SWD интерфейс.[9]

На фиг. 2.5 е представена блокова схема на процесорното ядро.



Фиг. 2.5 Блокова схема на микропроцесорно ядро „Cortex M7”

Микропроцесорните ядра от фамилията „Cortex M” се характеризират с два режима на работа:

- Режим на изпълнение на нишка – този режим се използва за изпълнение на апликационен код. Процесорът влиза в този режим веднага след включване;

- Режим на изпълнение на изключения – този режим се използва за изпълнение на прекъсвания и прекъсващи програми.

Софтуерът за тези процесори може да бъде разделен на две нива:

- Непривилегирован – софтуерът има ограничен достъп до някои от инструкциите, няма достъп до системния брояч, контролера на прекъсвания и системния контролен блок;

- Привилигиран – софтуерът има достъп до всички инструкции и може да използва всички ресурси.

Процесорите от „Cortex M” фамилията разполагат с 21 системни регистъра, като всички от тях са с дължина от 32 бита. Тези регистри са разделени на 3 основни групи:

- Регистри за основно ползване – 13 регистъра за основно ползване, като от първи до седми регистър включително са достъпни от всички инструкции, а от осми до дванадесети са достъпни от всички 32 битови инструкции и от някои 16 битови;

- Регистри със специално значение – 3 регистъра със специално значение. Регистър 13 се използва за указател към стековата памет. Регистър 14 се използва за запазване на адреса на връщане при прекъсвания. Регистър 15 се използва за запазване на текущия програмен адрес (програмен брояч).

- Регистри за специални цели – 5 регистъра за специални цели. Регистър 16 се използва за запазване състоянието на програмата, като той е разделен на 3 части;

- Апликационно състояние – съдържа системни флагове (Negative, Zero, Carry, Overflow, Q-Sticky saturation);

- Състояние на прекъсване ;

- Номер на прекъсване, изпълнявано в момента.

Регистър 17 се използва за позволяване и забраняване на отделни прекъсвания. Регистър 18 се използва за настройка на приоритета на прекъсванията. Регистър 19 се използва за ограничение на прекъсванията по приоритет. Регистър 20 се използва за контрол на стековата памет и нивото на достъп на софтуера.

Процесорите от „Cortex M” фамилията имат оптимизиран режим за бързо влизане и излизане от прекъсваща програма. Записът на системните регистри (Регистър 15 системен брояч, Регистър 16 състояние на програмата, Регистри за основно ползване от 0 до 3, Регистър 12, Регистър 14 адрес на връщане) се извършва автоматично от процесорното ядро, което позволява лесно използването на прекъсвания на програмния език „C”. „Cortex M3” и „Cortex M4” имат време за влизане и излизане от прекъсване 12 цикъла, а „Cortex M0” - 6 цикъла.[5,6,7,8,9]

Глава 3. РАЗРАБОТВАНЕ НА МОБИЛНА СИСТЕМА ЗА НАБЛЮДЕНИЕ

3.1 Блокова схема и обяснителна записка

Изходните данни и изискванията към мобилната система за наблюдение са следните:

- Използване на микроконтролер на фирмата „ST Microelectronics” за управление и събиране на данни от сензори;
- Използване на безжичен интерфейс за комуникация и управление на системата;
- Използване на камера за предаване на изображение в реално време.

На фиг.3.1. е показана блокова схема на системата за наблюдение.

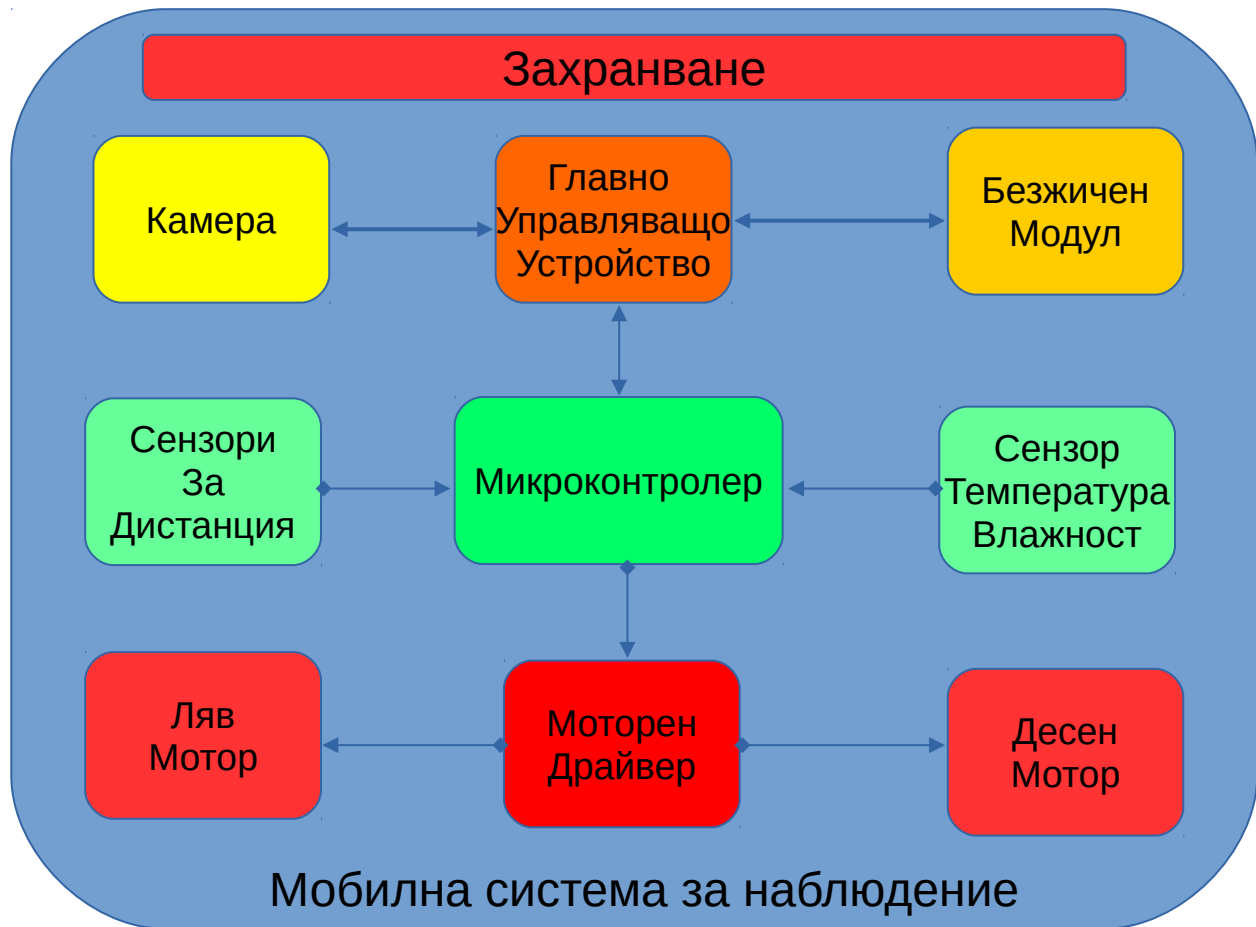
Проектираната мобилна система за наблюдение съдържа следните функционални блокове:

Главно управляващо устройство – функциите, които този блок изпълнява, са следните:

1) Осъществяване на безжична връзка с операторски компютър или мобилен телефон и обработка на данните за безжичната комуникация. Тази комуникация е разделена на два канала, показани на фиг. 3.1.

➤ Видео канал – този канал служи за изпращане на видео изображението от камерата в реално време.

➤ Управляващ канал – този канал служи за получаване на операторски команди и изпращане на данни от сензорите за дистанция, температура и влажност.



Фиг.3.1. Блокова схема на автоматизирана система за визуален контрол

2) Управление и четене на камера по сериен камера интерфейс версия 2 „CSI 2”. Физическия интерфейс за връзка с камерата е наречен „D-PHY” и има следните основни характеристики:

- Една двупроводна линия за подаване на тактов сигнал;
- Една или две двупроводни линии за подаване на информация;
- I2C интерфейс за управление;
- Максимална скорост на предаване на данни – 2.5 Gbps за канал [16].

3) Комуникация с микроконтролерния блок по сериен интерфейс – за осъществяване на комуникацията е използван I2C интерфейс, настроен на 100 kbps. Типа на комуникацията е господар – подчинен. Главното

управляващо устройство извършва ролята на господар и има възможността да изпраща команди за задаване скоростта на моторите или запитване за извличане на информация от сензорите за дистанция, температура и влажност.

Микроконтролер – този блок служи за управление на скоростта и посоката на въртене на два мотора чрез изпращане на управляващи сигнали към моторния драйвер. Също така този блок служи за събиране на данни от сензорите за дистанция, температура и влажност и се грижи за избягването на колизии и сблъсъци с други обекти при движение на мобилната система за наблюдение.

Сензори за дистанция - този блок се състои от сензори за измерване на дистанция и позволява адаптивен контрол и избягването на сблъсъци при движение на мобилната система. Основните параметри, по които се избират сензорите за дистанция, са следните:

- Обхват на ъгъла на измерване;
- Точност на измерване на дистанция;
- Максимално време за извършване на едно измерване;
- Минимална и максимална стойност на измерване.

Сензор температура-влажност – този блок осъществява измерване на температура и относителна влажност на околната среда, в която се намира мобилната система. Основните критерии за избиране на този блок са следните:

- Измервателен температурен обхват;
- Точност на измерване на температура;
- Измервателен обхват на относителна влажност;
- Точност на измерване на относителна влажност;

Моторен драйвер - този блок приема управляващите сигнали, подадени от микроконтролера и изработва нужните електрически сигнали

за въздействие върху моторите. Основните критерии, по които се избира този блок, са:

- Да се управлява с логически сигнали с ниво 3,3 V;
- Да осигурява нужното изходно напрежение и ток за задвижване на използваните мотори;
- Да съдържа защита от пренапрежение на входовете и изходите си;
- Да има малко време на закъснение спрямо входното напрежение и стръмни фронтове за нарастване и спадане на изходното напрежение.

Ляв и десен мотор - тези блокове са съставени от постояннотокови мотори, служещи за задвижване на мобилната система за наблюдение.

Камера – този блок служи за събиране на визуална информация от околната среда, в която се намира мобилната система за наблюдение и преобразуване на тази информация в RGB код.

3.2 Проектиране на модула за управление и модула за комуникация

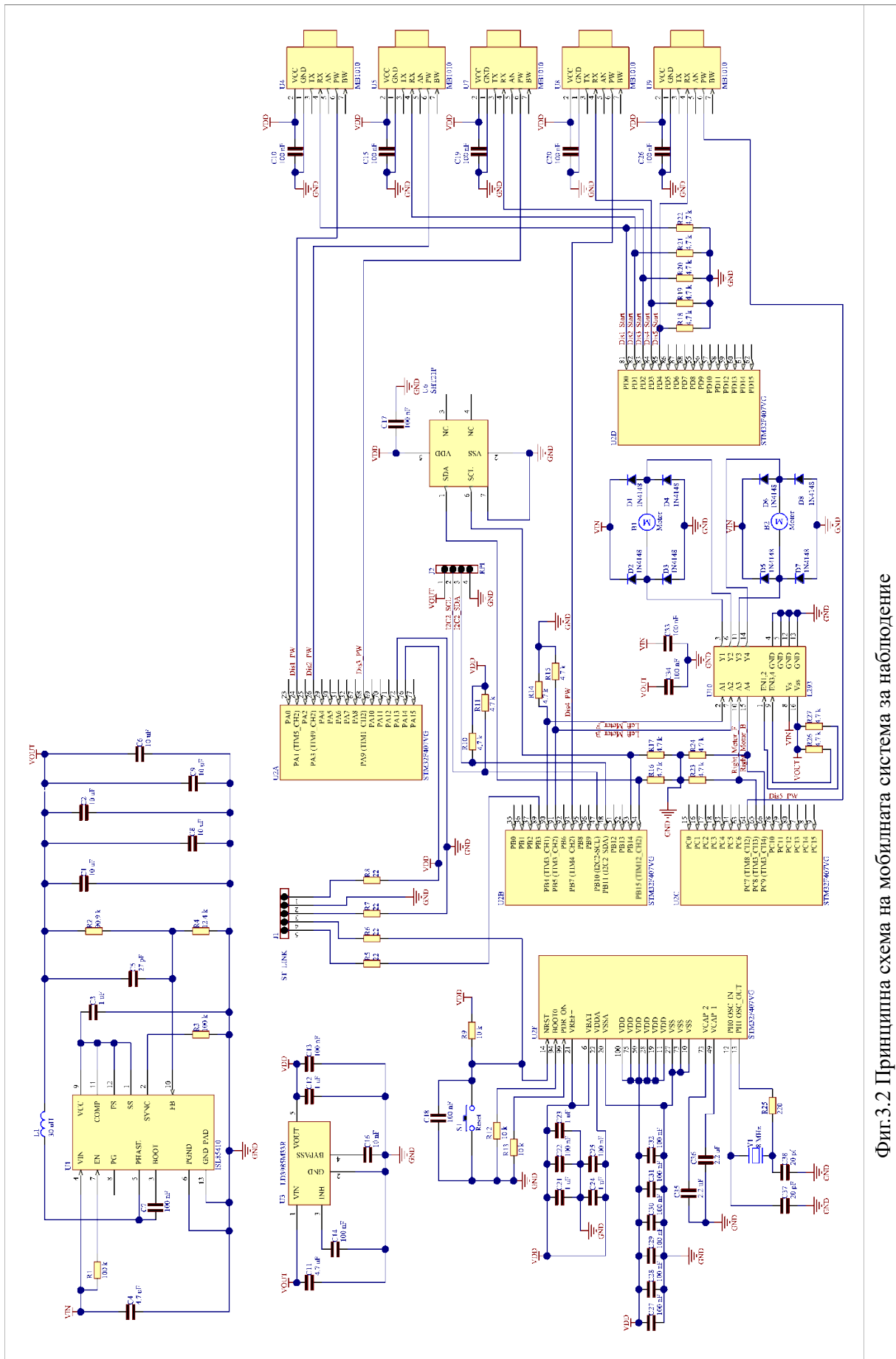
3.2.1 Принципна схема на мобилната система за наблюдение

На фиг.3.2 е представена проектираната принципна схема на мобилната система за наблюдение.

3.2.2 Разчет и съображения за избиране на елементите и техните стойности

Главното управляващо устройство е базирано на едноплатков компютър „Raspberry Pi 3“ със следните основни технически характеристики:

- Процесор 64 – битов „BCM2837“ с четири физически ядра, работещи с 1.2 GHz тактова честота;



Фиг.3.2 Принципна схема на мобилната система за наблюдение

- Оперативна памет 1 Gb;
- Безжични комуникационни интерфейси - WiFi модул BCM43143, Bluetooth модул;
- Жични комуникационни интерфейси – Ethernet, USB, UART, I2C, SPI, HDMI, CSI v.2.

Микроконтролерът, използван в системата за мобилно наблюдение, е STM32F407VG, произведен от фирмата „STMicroelectronics“. Той има следните основни технически характеристики:

- Ядро 32-битово ARM Cortex M4 с 1.25 DMIPS/MHz;
- Системен брояч, вграден в ядрото;
- Възможност за работа с плаваща и фиксирана запетая;
- Максимална тактова честота 168MHz;
- 1 Mbite програмна памет и 192 kbite оперативна памет;
- Комуникационни интерфейси – 3 x USART, 2x UART, 3x I2C, 3x SPI, 2x CAN, 2x USB, Ethernet;
- 12 брояча – 16-битови и два 32-битови;
- 3x 12-бита АЦП с максимална скорост на измерване 2.4MSPS;
- 2x 12-бита ЦАП;
- 2x контролер за директен достъп до паметта.

Таблица 3.1 визуализира използваните изводи от микроконтролера и тяхното предназначение в мобилната система за наблюдение.

Таблица 3.1 Използвани изводи от микроконтролера

Предназначение	Порт	Извод	Периферия
Комуникация с ГУУ	Порт В	10	I2C тактова честота
Комуникация с ГУУ	Порт В	11	I2C данни
Ляв мотор управление	Порт В	4	Брояч 3 изходен канал 1
Ляв мотор управление	Порт В	5	Брояч 3 изходен канал 2

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

Десен мотор управление	Порт С	8	Брояч 3 изходен канал 3
Десен мотор управление	Порт С	9	Брояч 3 изходен канал 4
Сензор за дистанция измерване	1 Порт А	1	Брояч 5 входен канал 2
Сензор за дистанция стартиране	1 Порт Д	0	Цифров изход
Сензор за дистанция 2 измерване	Порт А	3	Брояч 9 входен канал 2
Сензор за дистанция стартиране	2 Порт Д	1	Цифров изход
Сензор за дистанция 3 измерване	Порт А	9	Брояч 1 входен канал 2
Сензор за дистанция стартиране	3 Порт Д	2	Цифров изход
Сензор за дистанция 4 измерване	Порт В	7	Брояч 4 входен канал 2
Сензор за дистанция стартиране	4 Порт Д	3	Цифров изход
Сензор за дистанция 5 измерване	Порт С	7	Брояч 8 входен канал 2
Сензор за дистанция стартиране	5 Порт Д	4	Цифров изход
Сензор за температура и относителна влажност	Порт В	14	Цифров изход
Сензор за температура и относителна влажност	Порт В	15	Брояч 12 входен канал 2
Индикация при възникване на проблем	Порт Д	14	Цифров изход

За измерване на температура и влажност е избран сензор SHT21P, произведен от фирмата „Sensirion”, който представлява комбиниран сензор за измерване на температура в обхват от -40°C до 125°C и относителна влажност в обхват от 0 % до 100 % . Параметрите на сензора при измерване на температура са показани в таблица 3.2, а тези при измерване на влажност са показани в таблица 3.3.

Таблица 3.2 Параметри на сензора при измерване на температура

Параметър	Условие	Стойност	Мерна единица
Резолюция	14 бита	0.01	°C
Точност/толеранс	типична	±0.3	°C

Таблица 3.3 Параметри на сензора при измерване на относителна влажност

Параметър	Условие	Стойност	Мерна единица
Резолюция	12 бита	0.04	%RH
Точност/толеранс	типична	±2	%RH
Хистерезис		±1	%RH
Нелинейност		<0.1	%RH

Сензорът за температура и влажност разполага с един изход за данни, като информацията се представя във вид на широчинно-импулсна модулация (ШИМ) и един цифров вход, чрез който се указва каква информация да се изведе на изхода за данни. Задаване на логическа единица на цифровия вход означава извеждане на информация за относителна влажност, а логическа нула – температура. Изходът за данни на сензора е свързан към брояч на микроконтролера, който измерва коефициента на запълване и периода на изходния сигнал. След завършване на измерване микроконтролера изчислява температура (формула 3.1) и относителна влажност (формула 3.2)

$$T = -46.85 + 175.72 \cdot \frac{t_{PW}}{t_F} \quad (3.1)$$

$$RH = -6 + 125 \cdot \frac{t_{PW}}{t_F} \quad (3.2)$$

Където:

T – температура, [$^{\circ}\text{C}$];

t_{pw} – коефициент на запълване на изходния сигнал, [s];

t_f – период на изходния сигнал, [s];

RH – относителна влажност, [%];

Моторният драйвер е базиран на полумостова интегрална схема L293, произведена от фирмата „STMicroelectronics“, която има следните основни технически характеристики:

- Минимално входно напрежение, приемано за логическа единица 2.3V;

- Максималното изходно напрежение е 36 V;

- Максималния изходен ток на всеки изходен канал е 1 A;

- Максимален пиков ток на всеки изходен канал е 2 A;

- Изходният сигнал има малко времезакъснение спрямо входния сигнал и стръмни фронтове за нарастване и спадане. На таблица 3.4 са показани времевите характеристики на моторния драйвер;

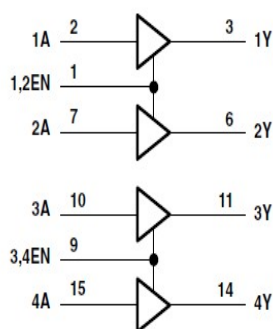
- Има възможност да управлява два постояннотокови мотора в двете посоки на въртене, което позволява ползването на един драйвер за задвижване на мобилната система за наблюдение.

Таблица 3.4 Времеви характеристики на моторния драйвер,

Символ	Параметър	Тестови условия	Типична стойност	Мерна единица
t_r	Нарастващ фронт	0.1 до 0.9 * $U_{изх}$	250	ns
t_f	Спадащ фронт	0.9 до 0.1 * $U_{изх}$	250	ns
T_{on}	Закъснение при включване	От 0.5* $U_{вх}$ до 0.5 * $U_{изх}$	700	ns
T_{off}	Закъснените при изключване	От 0.5* $U_{изх}$ до 0.5 * $U_{вх}$	200	ns

На фиг.3.3 е представена логическа диаграма на моторния драйвер L293D.

Изводи 1А и 2А се използват за управление на първия постоянен ток мотор, а изводи 3А и 4А за управление на втория. Извод 1,2EN се използва за разрешаване и забраняване на управлението на първия двигател, а извод 3,4EN за разрешаване и забраняване на управлението на втория двигател.



Фиг. 3.3 Логическа диаграма на моторния драйвер

Изводи 1А, 2А, 3А, 4А, 1,2EN и 3,4EN се управляват с логически сигнали, постъпващи от микроконтролера. Изводи 1Y и 2Y са изводите, към които се свързва и чрез които се задвижва първия постоянен ток мотор. Изводи 3Y и 4Y са с аналогична функция като 1Y и 2Y, но отговарят за втория постоянен ток мотор. На таблица 3.5 е представена таблицата на истинност на моторния драйвер.

Табл.3.5 Таблица на истинност моторния драйвер: Н-високо ниво; L-ниско ниво;

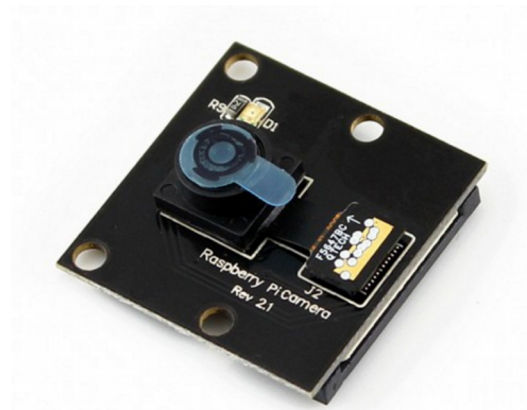
Z-високо импедансно състояние

Входно състояние на изводи 1,2EN и 3,4 EN	Входно състояние на изводи 1А,2А,3А,4А	Изходно състояние на изводи 1Y,2Y,3Y,4Y
Н	Н	Н
Н	L	L
L	Н	Z
L	L	Z

Избраната камера е Rpi Camera (D), произведена от фирмата „Waveshare”, която има следните основни параметри:

- OV5647 сензор с резолюция 5 мегапиксела;
- Размер на матрицата $\frac{1}{4}$ инч;
- Фиксиран фокус;
- Възможност за записване на видео със следните параметри: 1080p30 ; 720p60, 640x480p60/90;
- Размери 25мм x 24мм x 9мм;
- Стандартен сериен камера интерфейс версия 2.

На фиг. 3.4 е представено изображение на камерата, използвана в мобилната система за наблюдение. Избрана е поради компактните си размери, добрата резолюция и съвместимия интерфейс за комуникация с използваното главно управляващо устройство.



Фиг. 3.4 Камера използвана в мобилната система за наблюдение

Постояннотоковите мотори, използвани за задвижване на мобилната система за наблюдение, са със следните технически параметри:

- Препоръчително работно напрежение 6V;
- Максимално работно напрежение 12V;
- 240 оборота в минута след редуция при 6V напрежение;

- Съотношение на редуктура 1:48;
- Въртящ момент – 0.8 kg/cm.

На фиг. 3.5 е представено изображение на гумите, използвани за задвижване на мобилната система за наблюдение. Те са с диаметър 65 mm, което позволява достигането на скорост от 0,82 m/s при скорост на въртене на моторите 240 Rpm.



Фиг. 3.5 Гуми, използвани за задвижване на мобилната система за наблюдение

Сензорите за дистанция, избрани за мобилната система за наблюдение са „MB1010“, произведени от фирмата „Maxbotix“. Те представляват ултразвукови сензори за дистанция, които имат следните основни технически характеристики:

- Захранващо напрежение – 2.5V – 5V;
- Максимално време за измерване – 50 ms;
- Сериен интерфейс за комуникация – 9600 81N;
- Аналогов изход с изходен сигнал напрежение;
- Аналогов изход с импулсен изходен сигнал - $147\mu s = 25.4\text{ mm}$;
- Обхват на измерване – 0 до 6.45 m;
- Резолюция – 25.4 mm.

На фиг 3.6 е представено изображение на използвания сензор за дистанция MB1010. Изводите „Изпращане“ и „Получаване“ служат за осъществяване на серийна комуникация със сензора за дистанция. Изводът

„Аналогов изход“ извежда информация за дистанцията във вид на напрежение. Формулата, по която се изчислява дистанцията от напрежението, при захранващо напрежение 3.3 V, е следната (3.3):

$$d = \frac{V_a}{6.45 \text{ mV}} \times 25.4 \text{ [mm]} \quad (3.3)$$

Където:

d – дистанция, [mm];

V_a – напрежение на аналоговия изход.



Фиг. 3.6 Сензор за дистанция MB1010

Извода „Импулсен изход“ извежда информация за дистанция във вид на импулс с ширина, пропорционална на дистанцията. Формулата, по която се изчислява дистанцията, е следната (3.4):

$$d = \frac{t_{on}}{147 \text{ us}} \times 25.4 \text{ [mm]} \quad (3.4)$$

КЪДЕТО:

d – дистанция, [mm];

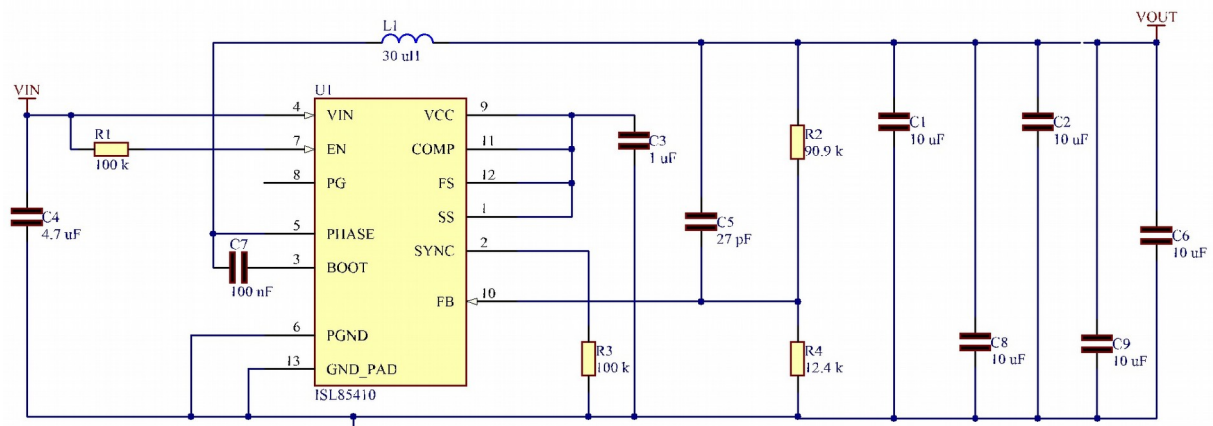
t_{on} – време в активно състояние.

За активиране на серийната комуникация „Контрол“ извода трябва да бъде установен в логическа нула. MB1010 има два режима на измерване - продължително (непрекъснато) и единично. Тези режими се контролират от логическото състояние на извода „Получаване“. При установяване на този извод за $20\mu s$ логическа единица се тригерира единично измерване, а за продължително (непрекъснато) измерване извода може да се свърже към захранване през повдигащ резистор.

Захранващият блок е базиран на интегрална схема ISL85410, произведена от фирмата „Intersil“, която представлява понижаващ преобразувател на постоянно напрежение в постоянно напрежение. Преобразувателят има следните основни технически характеристики:

- Диапазон на входното напрежение – 3 – 40V;
- Максимален изходен ток - 1A;
- Максимален пиков изходен ток – 1.7A;
- Максимална ефективност – 96%;
- Честота на превключване – 300 – 2000 KHz;
- Напрежение на обратната връзка – 600 mV;
- Регулируемо изходно напрежение.

Захранващото напрежение на главното управляващо устройство е 5V, поради което е избрано изходното напрежение на захранващия блок да бъде 5V. Схемата на преобразувателя на постоянно в постоянно напрежение е представена на фиг 3.7.



Фиг.3.7 Схема за преобразувателят на постоянно в постоянно напрежение

SS – този пин контролира времето за стартиране на преобразувателят чрез добавяне на кондензатор между този пин и земя. Ако SS бъде свързан към VCC, тогава времето за старт се определя от вътрешна верига и е със стойност 2 ms.

SYNC – чрез този пин се определя режима на работа на преобразувателя. При задаване на логическа единица, преобразувателят ще работи в режим на широчинно-импулсна модулация, а при установяване в 0 преобразувателя автоматично избира режима на работа (PWM или PFM).

Boot – към този пин се свързва кондензатор от 100 nF, който осигурява нужния заряд за включване на вградения N-канален MOSFET транзистор.

Vin - към този пин се подава входното напрежение.

PHASE - на този пин е подаден изводния сигнал от превключващите транзистори в преобразувателя и към него трябва да се свърже индуктивност. За изходно напрежение 5 V препоръчителната стойност на индуктивността е 30uH.

PGND - този пин се свързва директно към земя.

EN – този пин представлява вход, от който се стартира преобразувателя. При подаване на нула към този пин или свързване към

земя, преобразувателят се изключва, а при подаване на напрежение, по-високо от 1V, преобразувателят се включва.

VCC – този пин представлява изход от вграден в преобразувателя стабилизатор с напрежение на стабилизация 5V.

COMP – този пин служи за компенсация на грешката на вграден усилвател в преобразувателят. Когато този пин е свързан към VCC се използва вградена компенсация.

FS - чрез този пин се избира честотата на работа на преобразувателя, когато този пин е свързан към VCC, честотата на работа е 500 kHz.

GND - сигнална земя на преобразувателя.

FB - този пин служи за обратна връзка за регулиране. Напрежението на сравнение е 600 mV. Делителят на напрежение, реализиран от R2 и R4, осъществява обратна връзка от изходното напрежение. Формулата, по която той е изчислен, е следната (формула 3.5):

$$V_{fb} = \frac{R_4}{R_2 + R_4} \times V_{out} \quad (3.5)$$

Където:

V_{fb} – напрежение на обратната връзка 600 mV;

V_{out} – изходно напрежение 5V;

R4 – избран 12.4 kΩ

$$V_{fb} = \frac{R_4}{R_2 + R_4} \times V_{out} \rightarrow 0.6 \text{ V} = \frac{12.4 \text{ k}}{12.4 \text{ k} + R_2} \times 5 \text{ V} \rightarrow R_2 = \frac{5 \times 12.4 \text{ k} - 0.6 \times 12.4 \text{ k}}{0.6}$$

Полученото съпротивление за R2 е приблизително 90.933 kΩ, избира се стандартна стойност 90.9 kΩ 1%. След избиране на стойностите на резисторите напрежението на обратната връзка се получава:

$$V_{fb} = \frac{R_4}{R_2 + R_4} \times V_{out} \rightarrow V_{fb} = \frac{12.4 \text{ k}}{12.4 \text{ k} + 90.9 \text{ k}} \times 5 \text{ V} \rightarrow V_{fb} = 600.192 \text{ mV}$$

3.4 Алгоритъм на работа на мобилната система за наблюдение

3.4.1 Алгоритъм на работа на главното управляващо устройство

Избраният език за проектиране на софтуера на главното управляващо устройство е „C”. Както бе упоменото в т. 3.1, ГУУ има три основни задачи: безжична комуникация с операторски компютър или мобилен телефон, четене и управление на камера и комуникация с микроконтролер. Програмата на главното управляващо устройство е разделена на два процеса: основен процес и комуникационен процес. Подробно функционално описание на работата им е представено в т.3.4.1.2 (основен процес) и т.3.4.1.3 (комуникационен процес).

3.4.1.1 Безжична комуникация.

Безжичната комуникация с компютър или мобилен телефон се осъществява чрез Wifi. Видео каналът за изпращане на изображение в реално време трябва да бъде с минимален протоколен механизъм и минимално закъснение, поради което е избран User Datagram Protocol (UDP) протокол за комуникация с отдалеченото устройство. UDP протоколът използва лек модел за приемане и предаване с минимални протоколни механизми. Той не съдържа диалози за ръкостискане, също така няма гаранция за доставяне и няма защита чрез клониране. Чрез UDP протоколът връзката между два хоста се осъществява без предварителни съобщения, за създаване на специални предавателни канали или канали за данни. UDP е подходящ за цели, в които проверка за грешки и корекция или не е необходима, или се извършва от приложението, което води до намаляване на натоварването на ниво мрежов интерфейс. Времево-критични приложения често използват UDP, защото чакането за проверки и забавени пакети често не е подходяща опция в една система, работеща в реално време.[10,11]

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

Управляващ канал - съдържа информация, свързана с отчитания от сензорите контрол на скорост и посока на движение на мобилната система за наблюдение. Тази информация не е голяма по обем, но засяга пряко управлението на мобилната система, затова загуба на пакети с информация тук е недопустима. Поради този факт е избран Transmission Control Protocol (TCP) протокол, който осигурява сигурна, структурирана и проверена за грешка комуникация между устройствата. Структурата на TCP протокола е представена на таблица 3.6[12,13].

Таблица 3.6 Структура на TCP протокол за комуникация.

б а й т	0	1	2	3												
д у м т я	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31															
0 0	Порт номер на източника								Порт номер на дестинацията							
4 3 2	Номер на поредица															
8 6 4	Номер на потвърждение															
1 9 2 6	Офсет на данните	резерви рани	Флагове				Големина на прозорец за поучаване									
1 1 6 2 8	проверка								Спешен указател							
2 1 0 6 0	Опции															
... ..	Апликационни данни															

- Порт номер на източника – 16 битово число, съдържащо номера на порта на източника на съобщение;
- Порт номер на дестинацията – 16 битово число, съдържащо номера на порта на дестинацията на съобщението;
- Номер на поредица – 32 битово число, имащо двойна роля. Ако „SYN” флаг е установен в единица, то тогава това число е първоначалния

номер на поредицата. Ако „SYN” флагът е установен в нула, то тогава това число е акумулиран номер на поредица на първия байт от сегмент от текущата сесия;

➤ Номер на потвърждение – ако флаг „ACK” е установен в единица, то тогава в това 32 битово число се съдържа следващия номер на поредица, който получателят очаква;

➤ Офсет на данни – 4-битово число, което задава дължината на TCP протоколните данни в думи с дължина 32 бита;

➤ Резервирани – 3 бита резервирани за бъдещо ползване;

➤ Флагове – 9 бита, като всеки бит е различен флаг, отговарящ за определена настройка;

➤ Големина на прозореца за получаване – 16 битово число, което показва количеството съобщения (по подразбиране байтове), които изпращача на текущия сегмент очаква да получи;

➤ Проверка – 16 битово число, съдържащо сума за проверка на коректността на съобщението;

➤ Спешен указател – ако „URG” флагът е установен в единица, то тогава това 16 битово число е офсет на номера на поредицата, индикиращ последният спешен байт с потребителски данни;

➤ Опции – допълнителни опции, които могат да се включат в протоколовия механизъм;

➤ Апликационни данни – потребителски данни, които изпращачата апликация предоставя на получаващата апликация.[12,13]

3.4.1.2 Алгоритъм на работа на основния процес на програмата на главното управляващо устройство

Главното управляващо устройство работи с линукс базирана операционна система на име „Raspberian”, което позволява разделянето на програмата на отделни процеси. Основния процес на програмата на

главното управляващо устройство се грижи за стартиране и следене на състоянието на комуникацията с потребителското устройство. При стартиране на програмата главното управляващо устройство създава основния процес. След създаването му се преминава към изпълнението на програмата. На фиг. 3.8 е представена блокова диаграма на алгоритъма на основния процес.

1. Записване на команда за стартиране на видеоканал – записва се командата за стартиране на видеоканал в глобален низ;

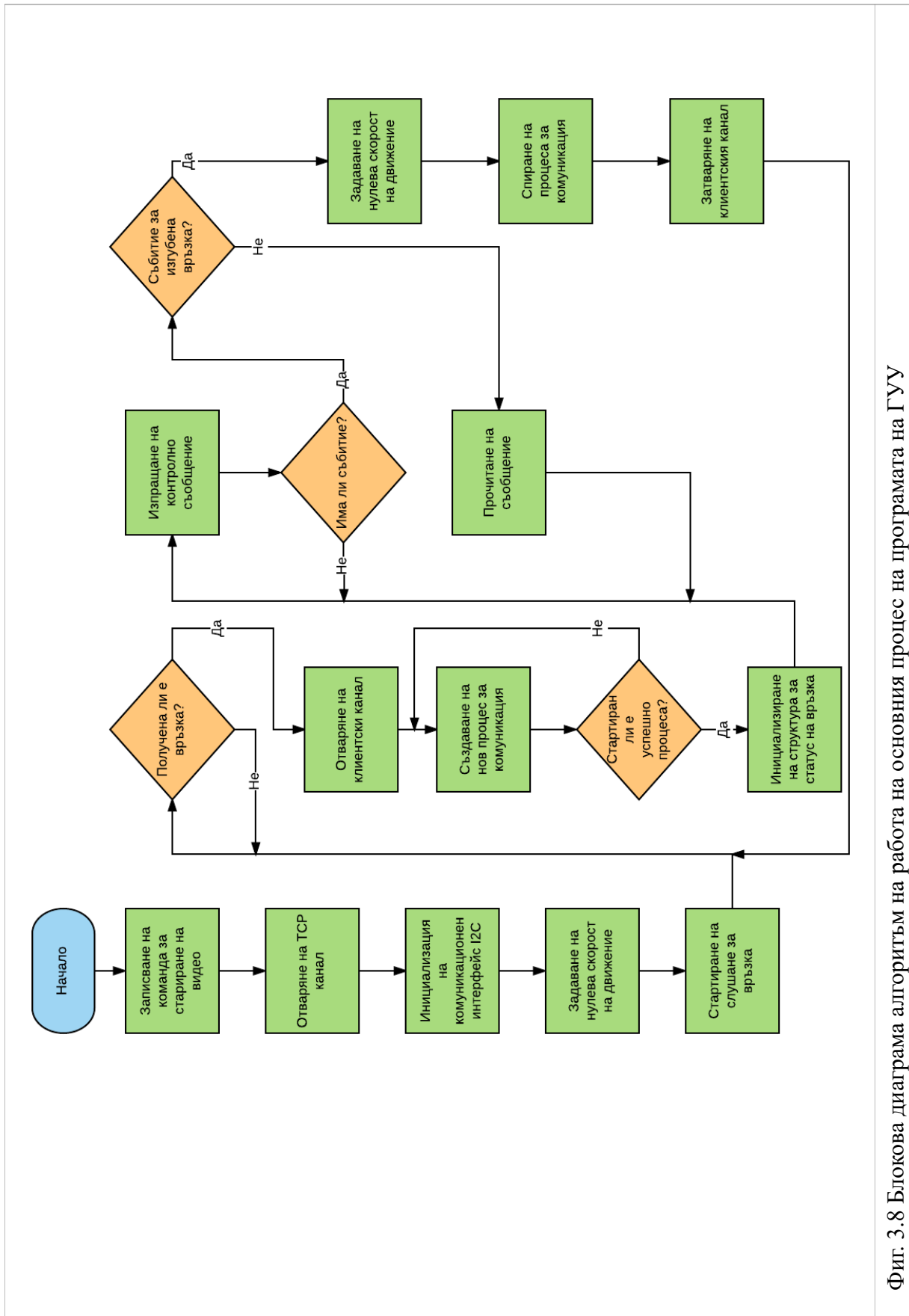
2. Отваряне на TCP канал – както бе упоменато в 3.1, безжичната комуникация с потребител е разделена на два канала: видеоканал и управляващ канал. Тук се създава управляващия канал, който се връзва към локален порт 5000 на ГУУ. След изпълнението на тази точка се преминава към точка 3;

3. Инициализация на комуникационен интерфейс I2C – инициализира се комуникационния интерфейс за връзка с микроконтролера. Настройва се стандартна скорост за обмен на данни 100 kbps и се стартира интерфейса. След изпълнението на тази точка се преминава към точка 4;

4. Задаване на нулева скорост на движение – изпраща се команда към микроконтролера за спиране на мобилната система за наблюдение. След изпълнението на тази точка се преминава към точка 5;

5. Стартиране на слушане за връзка – след извършване на първоначалните инициализации и конфигурации се позволява слушането за комуникация по вече създадения канал, след което се преминава към основния цикъл на програмата;

6. Получена ли е връзка – проверява се дали потребител се е свързал с главното управляващо устройство. При положителен резултат от проверката се продължава към точка 8, а при отрицателен резултат се извършва нова проверка;



Фиг. 3.8 Блокова диаграма алгоритъм на работа на основния процес на програмата на ГУУ

7. Отваряне на клиентски канал – след получаване с клиент се извършват протоколни механизми между потребителя и главното управляващо устройство. След завършване на тези механизми се позволява извършването на потребителска комуникация, което условно е наречено „Отваряне на клиентски канал“;

8. Създаване на процес за комуникация – след успешно отваряне на потребителски канал се създава нов процес, дъщерен на основния. Подробно обяснение за алгоритъма на работа на този процес е представено в т.3.4.1.3;

9. Стартиран ли е успешно процеса – извършва се проверка дали е успешно стартиран комуникационния процес. При положителен резултат от проверката се продължава към точка 11, а при отрицателен резултат се изпълнява точка 9 за нов опит за създаване на комуникационния процес;

10. Инициализация на структура за статус на връзка – проверката на състоянието на връзката се състои от няколко различни събития. Инициализацията на структура за статус на връзка се състои от избиране на кои събития да се проверяват. В конкретния случай са следните събития: събитие за чакащи данни за прочитане, събитие за чакащи данни с нисък приоритет за прочитане и събитие за прекратена връзка;

11. Изпращане на контролно съобщение – изпраща се контролно съобщение за проверка на състоянието на връзката;

12. Има ли събитие – извършва се проверка дали е възникнало събитие. При позитивен резултат от проверката се преминава към точка 14, а при негативен се изпълнява точка 12;

13. Събитие изгубена връзка – извършва се проверка дали възникналото събитие е за изгубена връзка. При позитивен резултат от проверката се преминава към точка 15. Негативен резултат от тази проверка означава че

има съобщение за прочитане в буфера за данни, затова се преминава към точка 18;

14. Задаване на нулева скорост на движение – изпраща се команда към микроконтролера за спиране на мобилната система за наблюдение. След изпълнението на тази точка се преминава към точка 16;

15. Спиране на процеса за комуникация – позитивния резултат от проверката за събитие изгубена връзка означава, че няма връзка с потребителя и процеса за комуникация ще бъде неактивен, затова той се спира и изтрива;

16. Затваряне на клиентски канал – затваря се клиентския канал, за да се освободи локалния порт на главното управляващо устройство и да е готов за приемане на нова връзка. След завършване на тази точка се преминава към точка 6 за проверка за нова връзка с потребител;

17. Прочитане на съобщение - прочита се съобщението от буфера за данни и се записва в локален буфер, използван от програмата. След изпълнение на тази точка се преминава към точка 12 за извършване на нова проверка за събитие.

3.4.1.3 Алгоритъм на работа на комуникационния процес на програмата на главното управляващо устройство

Комуникационният процес на програмата на главното управляващо устройство се грижи за обмена на данни между главното управляващо устройство и микроконтролера и между главното управляващо устройство и потребителското устройство. На фиг. 3.9 е представена блокова диаграма на алгоритъма на комуникационния процес. Стартирането и спирането на комуникационния процес се извършва от основния процес. Всеки път при осъществяване на връзка между потребителско устройство и главното управляващо устройство основния процес създава нов комуникационен процес, съдържащ параметрите на управляващия канал.

1. Получено ли е съобщение от потребител – извършва се проверка дали е получено съобщение от потребителско устройство. При наличие на съобщение се продължава към точка 2;

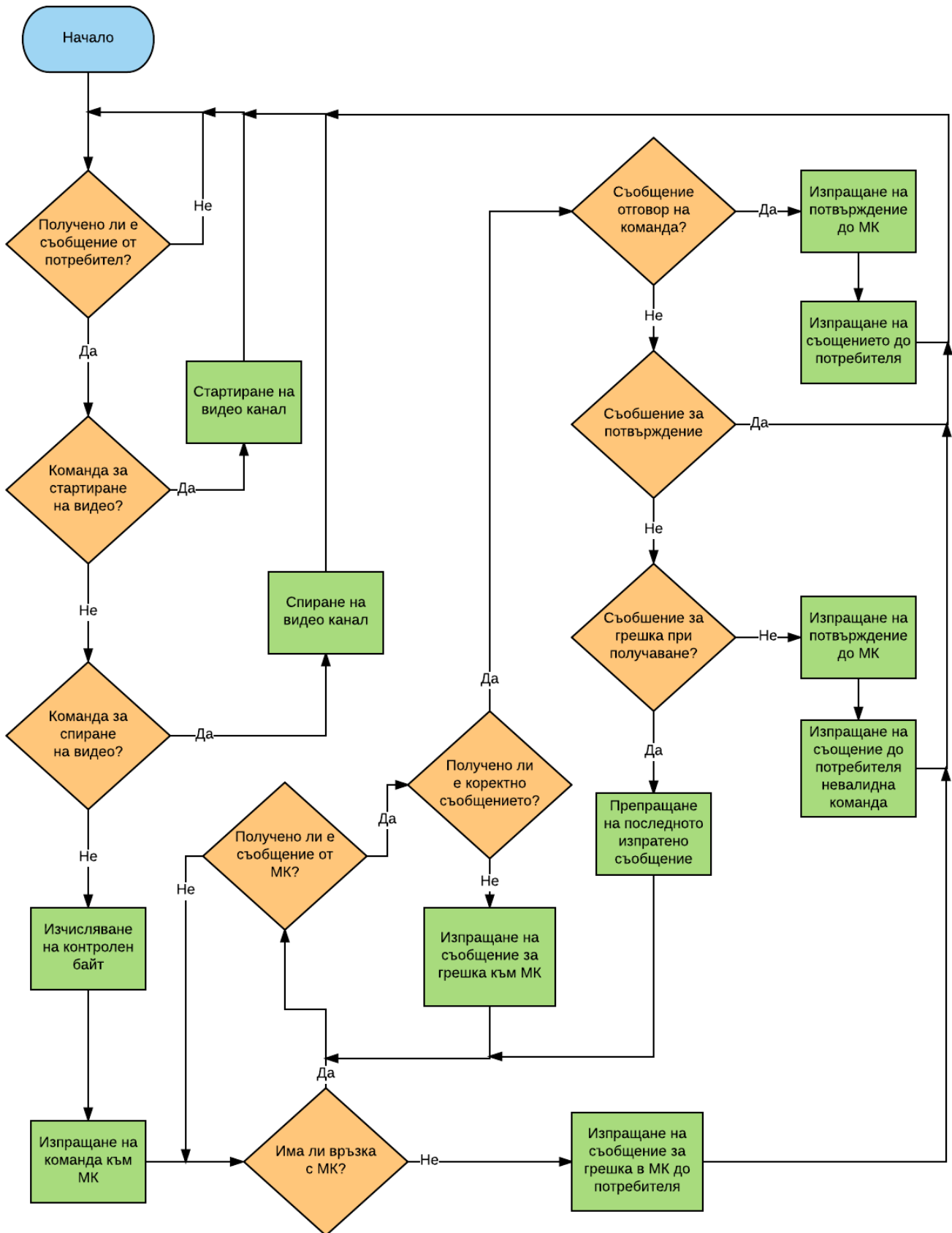
2. Команда за стартиране на видеоканал – тъй като комуникацията между главното управляващо устройство и потребителското устройство се извършва по TSP протокол, не е нужна проверка за коректно получено съобщение в програмата. Тази проверка е част от протоколния механизъм. След получаване на съобщение се проверява дали то е команда за стартиране на видео каналът. При положителен резултат от проверката се продължава към точка 3, а при отрицателен – точка 4;

3. Стартиране на видеоканал – стартира се видеоканалът и изпращането на кадри, заснети от камерата, към потребителското устройство. Изпращането на данни от камерата се извършва от отделен процес, който е независим от основния. След изпълнението на тази точка се преминава към точка 1 за изчакване на ново съобщение;

4. Команда спиране на видеоканал – извършва се проверка дали полученото съобщение е команда за спиране на видеоканала и изпращането на данни от камерата. При положителен резултат от проверката се продължава към точка 5, а при отрицателен – точка 6;

5. Спиране на видеоканал – спира се видеоканалът и изпращането на кадри, заснети от камерата, към потребителското устройство. Освобождава се локалният порт, използван за видеоканала и се изтрива задачата за четене и изпращане на данни от камерата. След изпълнението на тази точка се преминава към точка 1 за изчакване на ново съобщение;

6. Изчисляване на контролен байт – ако полученото съобщение не е команда за стартиране или спиране на видеоканалът, то тогава тя се отнася или за задаване скорост и посока на движение, или за прочитане на данни от сензорите. Командите за движение и прочитане на данни от сензорите са



Фиг. 3.9 Блокова диаграма на алгоритъма на работа на комуникационния процес

същите, както в протокола за комуникация между главното управляващо устройство и микроконтролера, поради което е възможно полученото съобщение от потребителското устройство да бъде пратено към микроконтролера след добавяне на контролен байт. Извършва се операция сума по модул 2 на байтовете от съобщението, в резултат на което се получава стойността на контролния байт. След изпълнение на тази точка се преминава към точка 7;

7. Изпращане на команда към микроконтролера – след изчисление на контролния байт се изпраща съобщението към микроконтролера;

8. Има ли връзка с микроконтролер – проверява се дали е налична комуникация с микроконтролера. Това се осъществява чрез изчакване на определен интервал от време за отговор. При изтичане на това време без получаване на отговор от микроконтролера се приема, че има проблем при него. Тук се проверява дали този интервал от време е изтекъл, ако не е изтекъл се продължава към точка 10, а ако е изтекъл – точка 9;

9. Изпращане на грешка в микроконтролера до потребителя – изпраща се съобщение за грешка в микроконтролера до потребителското устройство, след което се преминава към точка 1 за изчакване на ново съобщение от потребителското устройство;

10. Получено ли е съобщение от микроконтролера – проверява се дали е получено съобщение от микроконтролера. Ако е получено съобщение се продължава към точка 11, ако не – точка 8;

11. Получено ли е коректно съобщението - проверява се дали съобщението е получено коректно като се изпълнява операция сума по модул 2 на всички байтове от съобщението (от байт 0 до байт 4 включително). Резултатът се сравнява с байт 5 от съобщението. При позитивен резултат от сравнението програмното изпълнение продължава към точка 13, а при негативен – към точка 12;

12. Изпращане на съобщение за грешка - изпраща се съобщение за грешка при проверка за коректност на съобщението. След завършване на изпращането се преминава към точка 10 за изчакване на повторно изпращане на съобщението от микроконтролера;

13. Съобщение отговор на команда – извършва се проверка дали полученото съобщение от микроконтролера е отговор на команда за задаване на посока и скорост на движение или команда за извличане на информация от някой от сензорите. При положителен резултат от проверката се преминава към точка 14, а при отрицателен – точка 16;

14. Изпращане на потвърждение до микроконтролера – изпраща се потвърждение до микроконтролера, че съобщението е получено коректно, след което се продължава към точка 15;

15. Изпращане на съобщението до потребителя – премахва се контролния байт от съобщението, след което то се изпраща към потребителското устройство;

16. Съобщение за потвърждение – ако полученото съобщение от микроконтролера не е отговор на команда се проверява дали е съобщение за потвърждение. При позитивен резултат от проверката се продължава към точка 1 за изчакване на нова команда от потребителското устройство, а при негативен – точка 17;

17. Съобщение за грешка при получаване – проверява се дали полученото съобщение от микроконтролера не е за грешка при получаване (несъответствие между изпратеният от ГУУ контролен байт и изчисления от микроконтролера). При позитивен резултат от тази проверка се продължава към точка 18 , а при негативен – точка 19;

18. Препращане на последното изпратено съобщение - позитивният резултат от проверка за съобщение за грешка при получаване означава, че микроконтролера не е получил коректно последното изпратено съобщение от главното управляващо устройство, затова то се препраща на ново. След

изпълнение на тази точка се преминава към точка 10 за изчакване на отговор от микроконтролера;

19. Изпращане на потвърждение до микроконтролера – изпраща се потвърждение до микроконтролера, че съобщението е получено коректно, след което се продължава към точка 20;

20. Изпращане на съобщение до „невалидна команда“ потребителя – ако се стигне до тази част от програмата означава, че микроконтролерът е изпратил съобщение за непозната (невалидна) команда. Изпраща се съобщение до потребителското устройство, че изпратената команда не се поддържа от системата.

3.4.2 Комуникация между главно управляващо устройство и микроконтролер

Комуникацията между главното управляващо устройство и микроконтролера, както бе упоменато в т.3.2, се осъществява по I2C интерфейс. Този интерфейс представлява синхронен сериен интерфейс за комуникация с две двупосочни отворен дрейн линии. Едната линия служи за задаване на тактова честота и се нарича „Serial Clock Line” (SCL), а другата се използва за предаване на данни и се нарича „Serial Data Line” (SDA). Стандартно адресирането по този интерфейс е седем или десет битово. Устройствата, комуникиращи чрез този интерфейс, могат да използват следните четири режима на работа:

- Господар изпращане – командващото устройство изпраща данни към подчинено;
- Господар получаване – командващо устройство получава данни от подчинено;
- Подчинен изпращане – подчиненото устройство изпраща данни към командващото;

● Подчинен получаване – подчиненото устройство получава данни от командващото[14,15].

Използваният протокол за комуникация между главното управляващо устройство и микроконтролерът има фиксирана дължина на съобщение от 6 байта. Първият байт е команден байт. Байтовете от втори до пети са байтове за данни, а последният байт служи за проверка за коректност на съобщението чрез изпълнение на операция сума по модул две на предходните 5 байта. В таблица 3.7 са представени командите и посоката на данни на използване протокол.

Таблица 3.7 Подържани команди и посока на данни в използвания протокол

ГУУ – главно управляващо устройство, МК - микроконтролер

Команден байт	Байтове за данни				Посока на комуникация	Описание	
	Байт 1	Байт 2	Байт 3	Байт 4			Байт 5
0x00	0	0	0	0	0xAA	-	Потвърждение за получено коректно съобщение
0x01	Посока на движение	Линейна скорост	0	0	Ъглова скорост	ГУУ → МК	Тази команда служи за задаване на движение
0x01	Посока на движение				0xFF (отказ)	МК → ГУУ	Отказ за движение в конкретна посока
0x02	0	0	0	0	0	ГУУ → МК	Команда за прочитане данни за температура и относителна влажност
0x02	Температура	Температура	Влажност	Влажност	Влажност	МК → ГУУ	Отговор, съдържащ данни за температура и относителна влажност

0x03	0	0	0	Номер на сензор за дистанция	ГУУ → МК	Команда за прочитане на данни от конкретен сензор за дистанция
0x03	Номер на сензора за дистанция (0xFF невалиден сензор)	Дистанция	Дистанция	Дистанция	МК → ГУУ	Отговор, съдържащ данни за дистанция от конкретен сензор
0xF0	0	0	Номер на команда	0xFF	МК → ГУУ	Отговор за валидна команда
0xFE	0xFF Проблем при сензор за температура	0xFF Проблем при сензор за влажност	0xFF Проблем при сензор за дистанция	Номер на сензора за дистанция	МК → ГУУ	Отговор при проблем с някой от сензорите
0xFF	0	0	0	0	-	Грешка в проверка за коректност на съобщението

3.4.3 Алгоритъм на работа на микроконтролера

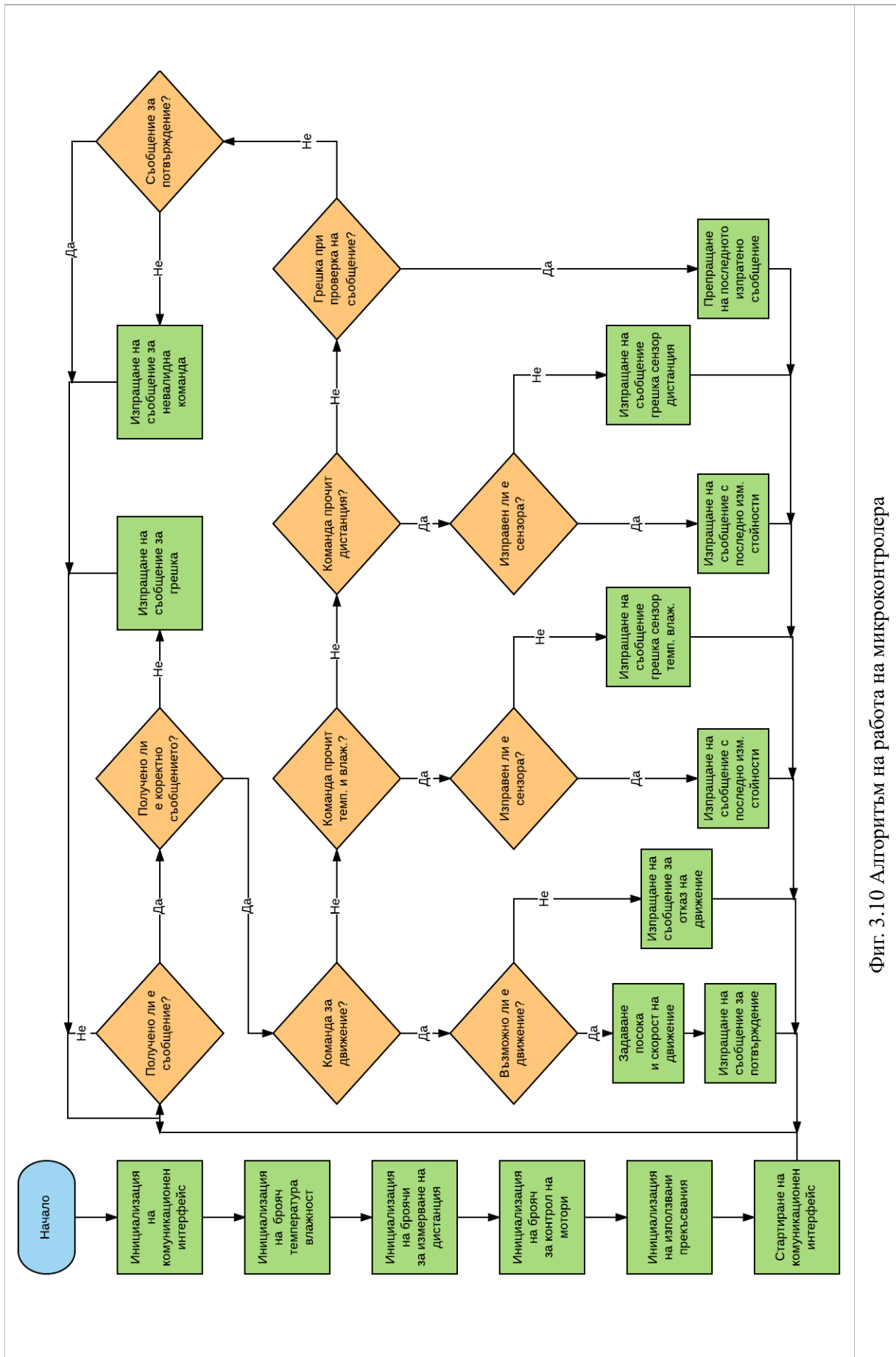
Микроконтролерът, както бе упоменато в т.3.1, трябва да управлява движението на системата, да събира данни и обработва информацията от сензорите за температура, влажност и дистанция и да следи разстоянието между мобилната система за наблюдение и други обекти, на базата на което да взема решения за предотвратяване на сблъсъци. На фиг. 3.10 е представена блокова диаграма на алгоритъма на работа на главната програма на микроконтролера.

1. Инициализация на комуникационния интерфейс – извършва се настройка на комуникационния интерфейс за комуникация с главното управляващо устройство и настройка на изводите, използвани от него. Задава се скорост на трансфер 100 kbps на използвания интерфейс I2C2;

2. Инициализация на брояч за измерване на температура и влажност – сензорът за температура и влажност генерира изходен сигнал широчинно-импулсна модулация, поради което е използван брояч, настроен в режим за измерване на ШИМ. В този режим са нужни два входни канала от брояча, които да са дадени на късо. Броячите в STM32F407 имат програмна настройка за включване на входен канал 1 и входен канал 2 на късо. При измерването на сигнала се използват 2 регистъра за задържане и сравнение. Броячът се настройва да се рестартира всеки път при постъпване на нарастващ фронт на входният сигнал. Регистър за запазване 1 записва стойността, до която е отброил брояча при постъпване на падащ фронт на входния сигнал, а регистър за запазване 2 – стойността на брояча при нарастващ фронт на сигнала преди рестартиране на брояча. За да се изчисли коефициента на запълване на широчинно-импулсната модулация, се разделя стойността от регистър за запазване 1 на стойността от регистър за запазване 2;

3. Инициализация на броячи за измерване на дистанция – за измерването на дистанция от MB1010 е избран да се използва импулсния изход. Тъй като сензорите трябва да се стартират поотделно за избягване на взаимни смущения, не може да се използва непрекъснат режим на работа. Броячите се настройват в режим за измерване на ШИМ, както при сензора за температура и влажност, с тази разлика, че регистър за запазване 2 не се използва, тъй като периода на сигнала не е необходим. Допълнително за измерването на дистанция се използват изводи, настроени като цифрови изходи за тригериране на сензорите за дистанция;

4. Инициализация на брояч за контрол на мотори - Моторите се управляват изцяло от брояч 3, както е посочено в таблица 3.1. Броячът е настроен в режим на генериране на широчинно-импулсна модулация, като изходен канал 1 и изходен канал 2 управляват левият мотор, а изходен



Фиг. 3.10 Алгоритъм на работа на микроконтролера

канал 3 и изходен канал 4 – десния. Неактивното състояние на каналите е логическа нула, а активното е логическа единица. За задвижване на мобилната система за наблюдение напред на изходен канал 1 и изходен канал 3 трябва да се подаде сигнал с коефициент на запълване по-голям от 0, а на канал 2 и канал 4 сигналът трябва да бъде с коефициент на запълване 0 (задаване на неактивно състояние). За коректно управление канал 1 и канал 2 не трябва да бъдат установявани в активно състояние едновременно, което се отнася и за канал 3 и канал 4. Регулиране на скоростта на завиване се извършва чрез промяна разликата между коефициентите на запълване на каналите, управляващи левия мотор спрямо каналите, управляващи десния канал;

5. Инициализация на използваните прекъсвания – след настройка на използваната периферия и конфигурация на изводите на микроконтролера следва инициализация на прекъсванията, които ще бъдат използвани в програмата. Позволяват се прекъсвания от избраните броячи за измерване на температура, влажност и дистанция. При постъпване на прекъсване от някой от тези броячи се извиква подпрограма за изчисляване на съответния параметър. Позволява се прекъсване през 1 ms от системния брояч, вграден в ядрото на микроконтролера. Този брояч се използва за времева синхронизация между задачи и откриване на проблеми в сензорите за температура, влажност и дистанция. Откриването на проблем в сензорите се извършва чрез проверка на времето, през което не е получавана информация от тях (ако това време е повече от предварително зададена стойност програмата приема, че сензорът е повреден и установява, в единица , флаг отговарящ за съответния сензор);

6. Стартиране на комуникационния интерфейс – след завършването на всички инициализационни процеси се позволява комуникационния интерфейс за връзка с главното управляващо устройство;

7. Получено ли е съобщение - след стартиране на комуникационния интерфейс се влиза в основния цикъл на програмата. Изчаква се получаване на съобщение от главното управляващо устройство. При получаване на цяло съобщение (6 байта) се преминава към точка 8;

8. Получено ли е коректно съобщението - проверява се дали съобщението е получено коректно като се изпълнява операция сума по модул 2 на всички байтове от съобщението (от байт 0 до байт 4 включително). Резултатът се сравнява с байт 5 от съобщението. При позитивен резултат от сравнението програмното изпълнение продължава към точка 10, а при негативен – към точка 9;

9. Изпращане на съобщение за грешка - изпраща се съобщение за грешка при проверка за коректност на съобщението. След завършване на изпращането се преминава към точка 7 за изчакване на повторно изпращане на командата от главното управляващо устройство;

10. Команда за движение – при коректно пристигнало съобщение се проверява дали съобщението е за команда за движение, ако е така се продължава към точка 11, ако не – точка 15;

11. Възможно ли е движението – проверява се зададената посока на движение и последно отчетените данни от сензорите за дистанция. Преценява се дали е възможно движението в зададената посока, като се провери дали стойността на сензорите, отговарящи за конкретната посока, са по-малки от предварително зададена гранична стойност. Ако стойностите са по-малки, се продължава към точка 12, ако не – точка 13;

12. Изпращане на съобщение за отказ на движение – изпраща се съобщение за отказ за движение в зададената посока поради прекалено къса дистанция до обект. След изпълнение на този точка се преминава към точка 7 за изчакване на ново съобщение;

13. Задаване посока и скорост на движение – след положително премината проверка за възможност на движение се изчисляват и задават

коэффициентите на запълване на сигналите, управляващи моторите, за да се установи желаната посока и скорост на движение;

14. Изпращане на съобщение за потвърждение – след изпълнението на точка 13 се изпраща съобщение за потвърждение, че командата е приета и изпълнена коректно. След изпълнение на тази точка се преминава към точка 7 за изчакване на ново съобщение;

15. Команда прочит на температура и влажност – ако получената команда не е за движение, се проверява дали е за прочит на сензора за температура и относителна влажност. При положителен резултат от проверката се преминава към точка 16, а при отрицателен – точка 19;

16. Изправен ли е сензора – проверява се дали сензора е изправен чрез проверка на програмен флаг, отговарящ за този сензор. Ако флагът е установен в 1, то сензорът работи коректно и се преминава към точка 18, ако флагът е установен в 0 - точка 17;

17. Изпращане съобщение за грешка сензора за температура и влажност – след установяване на неизправност на сензора за температура и влажност се изпраща съобщение към главното управляващо устройство за проблем в сензора. След изпълнение на тази точка се преминава към точка 7 за изчакване на ново съобщение;

18. Изпращане на съобщение с последно измерените стойности – при положителен резултат от проверката за изправност на сензора за температура и влажност се изпраща съобщение, съдържащо последно измерените стойности на параметрите. След изпълнение на тази точка се преминава към точка 7 за изчакване на ново съобщение;

19. Команда прочит дистанция – при негативен резултат от проверката за команда за прочит на температура и влажност се проверява дали получената команда е за прочит на дистанция от конкретен сензор. При позитивен резултат от тази проверка се преминава към точка 20, а при негативен – точка 23;

20. Изправен ли е сензора – първо се извършва проверка от кой сензор за дистанция се изисква информация и след това се проверява дали сензора е изправен чрез проверка на програмен флаг, отговарящ за съответния сензор. Ако флагът е установен в 1, то сензорът работи коректно и се преминава към точка 22, ако флагът е установен в 0 - точка 21;

21. Изпращане съобщение грешка при сензор за дистанция - след установяване на неизправност на съответния сензор за дистанция се изпраща съобщение към главното управляващо устройство за проблем в сензора. След изпълнение на тази точка се преминава към точка 7 за изчакване на ново съобщение;

22. Изпращане на съобщение с последно измерените стойности – при положителен резултат от проверката за изправност на съответния сензор за дистанция се изпраща съобщение, съдържащо номера на сензора и последно измерената стойност от него. След изпълнение на тази точка се преминава към точка 7 за изчакване на ново съобщение;

23. Грешка при проверка на съобщение - при негативен резултат от проверката за команда за прочит на дистанция се проверява дали получената команда е за грешка при получаване на съобщение. При позитивен резултат от тази проверка се преминава към точка 24, а при негативен – точка 25;

24. Препращане на последното изпратено съобщение – позитивния резултат от проверка за команда за грешка при проверка на съобщение означава, че главното управляващо устройство не е получило коректно (несъответствие между изпратения резултат от операция сума по модул 2 и изчисления) последното изпратено съобщение от микроконтролера, затова то се препраща наново. След изпълнение на тази точка се преминава към точка 7;

25. Съобщение за потвърждение - при негативен резултат от проверката за команда за грешка при проверка на съобщение се проверява дали

получената команда е съобщение за потвърждение. Позитивен резултат от тази проверка означава, че изпратения отговор от микроконтролера на съответна команда е приет успешно и изпълнението на програмата продължава от точка 7, а негативния резултат означава, че е получено коректно съобщение, но командата е непозната и програмата продължава към точка 26;

26. Изпращане на съобщение за невалидна команда – изпраща се съобщение към главното управляващо устройство, че получената команда е невалидна и системата не я познава. След изпълнение на тази точка се продължава към точка 7.

3.4.4 Алгоритъм на работа на потребителското устройство.

Потребителското устройство представлява мобилен телефон, работещ с Андроид операционна система. Програмата е написана на Андроид и се разделя на 3 основни части:

- свързваща част – подпрограма, използвана за осъществяване на връзка между главното управляващо устройство и потребителското устройство;
- управляваща част – подпрограма, използвана за управление на движението на мобилната система за наблюдение и визуализиране на видео, изпратено от главното управляващо устройство;
- измерваща част – подпрограма, осъществяваща четене на информация от сензорите на системата и визуализиране на данните на дисплея на потребителското устройство.

3.4.4.1 Алгоритъм на работа на подпрограмата за свързване

На фиг 3.11 е представена блок диаграма на алгоритъма на работа на свързващата подпрограма.

1. Визуализиране на графичен дизайн – след стартиране на програмата се извършва визуализация на графичния дизайн, който включва задаване на фон на дисплея и разчертаване на бутони за връзка и за сканиране;

2. Инициализация на бутони – след създаването на графичното изображение на бутоните се извършва инициализация на функцията на всеки бутон;

3. Натиснат ли е бутон – извършва се проверка дали е натиснат бутон. При позитивен резултат от тази проверка се продължава към точка 4 за установяване на кой бутон е натиснат, а при негативен резултат се извършва нова проверка за натиск на бутон;

4. Бутон за връзка – извършва се проверка дали е натиснат бутон за връзка „Connect”. При позитивен резултат от тази проверка се продължава към точка 5, а при негативен – точка 18;

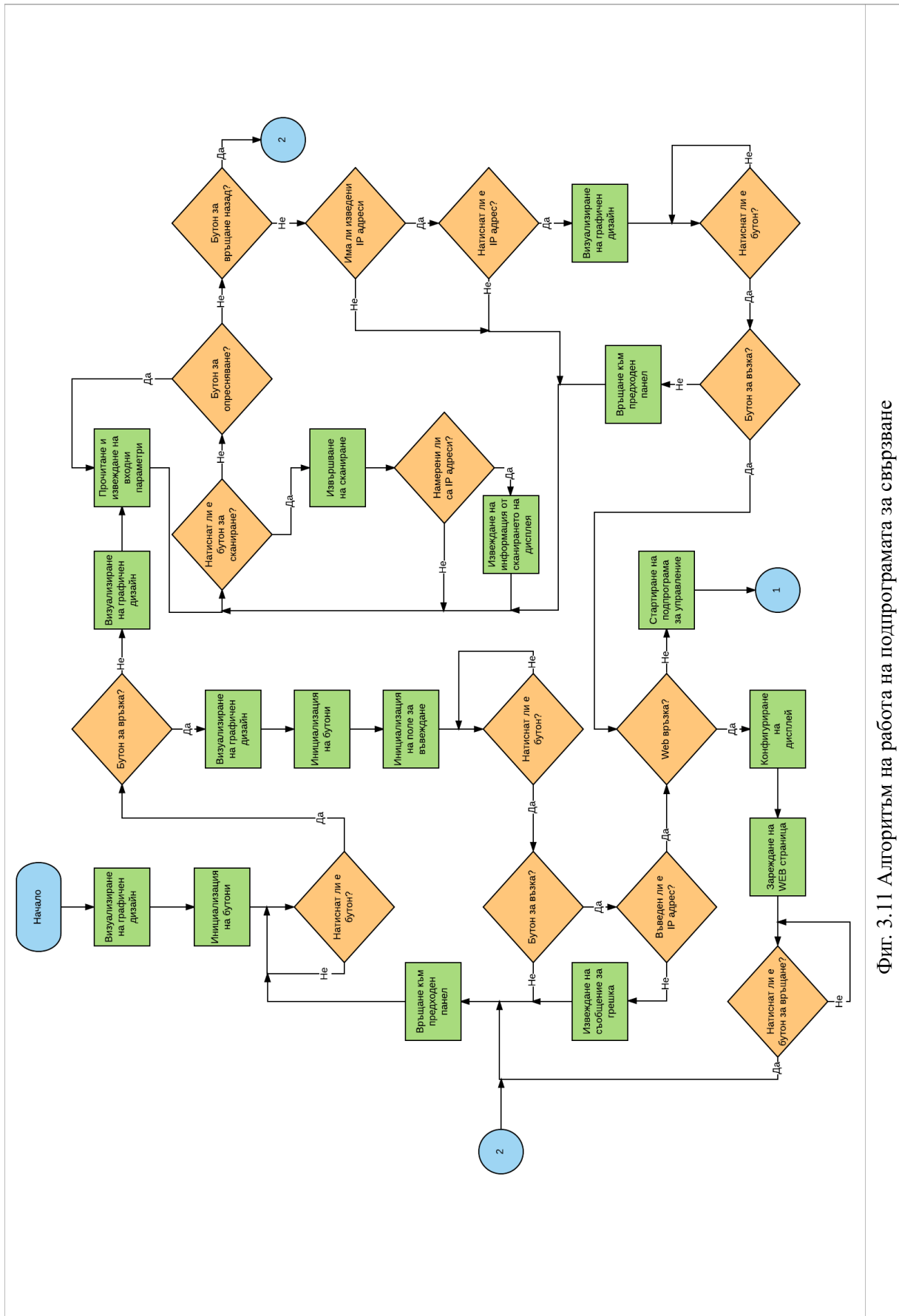
5. Визуализиране на графичен дизайн – изчиства се дисплея и се изчертава панел за връзка, съдържащ бутони за връзка и отказ, поле за въвеждане на IP адрес, опция за избиране на WEB или канална връзка;

6. Инициализация на бутони – инициализират се функциите на бутоните;

7. Инициализация на поле за въвеждане – инициализира се полето за въвеждане и се дава достъп на потребителя да въвежда символи от клавиатура в него;

8. Натиснат ли е бутон - извършва се проверка дали е натиснат бутон. При позитивен резултат от тази проверка се продължава към точка 9 за установяване на кой бутон е натиснат, а при негативен резултат се извършва нова проверка за натиск на бутон;

9. Бутон за връзка – извършва се проверка дали е натиснат бутон за връзка. При позитивен резултат от тази проверка се продължава към точка 11. Негативен резултат от проверката означава, че е натиснат бутон за отказ, затова се продължава към точка 10;



Фиг. 3.11 Алгоритъм на работа на подпрограма за свързване

10. Връщане на предходен панел - изчиства се дисплея на потребителското устройство и се изчертава входния панел. След завършване на тази операция се продължава към точка 3;

11. Въведен ли е IP адрес – извършва се проверка дали е въведен IP адрес за връзка на полето за въвеждане. При позитивен резултат е продължава към точка 13, а при негативен – точка 12;

12. Извеждане на съобщение за грешка – извежда се съобщение на дисплея за грешка при опит за връзка. Съобщението гласи, че не е въведен IP адрес за връзка. След изпълнението на тази точка се продължава към точка 10;

13. Web връзка – както бе упоменато в точка 5, има опция за избиране на типа връзка. Тук се извършва проверка дали е избрана Web връзка. При позитивен резултат от проверката се продължава към точка 15, а при негативен - точка 14;

14. Стартиране на подпрограма за управление – негативния резултат от проверката за Web връзка означава че е избрана канална връзка. Както бе упоменато в т.3.4.1.1, мобилната система за наблюдение използва управляващ канал, реализиран чрез TCP протокол, затова се стартира подпрограмата за управление, която използва този канал за управление на системата. Алгоритъма на работа на подпрограмата за управление е подробно обяснен в т.3.4.4.2;

15. Конфигурация на дисплей - изчиства се дисплея на потребителското устройство и се изчертава панел за web връзка;

16. Зареждане на web страница – извършва се опит за web връзка с въведения от потребителя IP адрес. При неуспех се извежда съобщение за грешка при свързване. Този тип връзка е добавен, тъй като е предвидено в бъдеще добавяне на управление на мобилната система за наблюдение през web страница;

17. Натиснат ли е бутон за връщане – проверява се дали е натиснат бутон за връщане назад. При позитивен резултат от тази проверка програмното изпълнение продължава от точка 10;

18. Визуализация на графичен дизайн – негативен резултат от проверката, извършена в точка 4, означава че е натиснат бутон за сканиране. Изчиства се дисплея на потребителското устройство и се изчертава панел за сканиране, съдържащ бутон за сканиране, бутон за опресняване, поле за изписване на IP адрес;

19. Прочитане и извеждане на входни параметри – проверява се дали е налична безжична локална връзка и IP адреса на потребителското устройство. Проверената информация се извежда на дисплея;

20. Натиснат ли е бутон за сканиране – проверява се дали е натиснат бутон за стартиране на сканиране. При позитивен резултат от тази проверка се продължава към точка 24, а при негативен – точка 21;

21. Извършване на сканиране – извършва се сканиране за устройства в безжичната локална мрежа. Сканирането се извършва чрез изпращане на контролно съобщение до IP адреси, завършващи на номера от 1 до 255. При отговор на контролното съобщение се запазва IP адреса на устройството, от което е получен отговора;

22. Намерени ли са IP адреси – извършва се проверка дали са намерени устройства в безжичната локална мрежа. При позитивен резултат от проверката се продължава към точка 23, а при негативен – точка 20;

23. Извеждане на информация от сканирането на дисплея – позитивния резултат от проверка за намерени IP адреси означава, че едно или повече устройства са отговорили на контролното съобщение. Изписват се на дисплея IP адресите на тези устройства;

24. Бутон за опресняване – проверява се дали е натиснат бутон за опресняване на входните параметри. Позитивен резултат от тази проверка означава, че потребителя желае да опреси изведените входни данни на

дисплея, затова се продължава към точка 19. При негативен резултат от проверката се продължава към точка 25;

25. Бутон за връщане назад – извършва се проверка дали е натиснат бутон за връщане към предходния панел. При позитивен резултат от тази проверка се продължава към точка 10, а при негативен – точка 26;

26. Има ли IP адреси – проверява се дали има изведени IP адреси на устройства, вързани към безжичната локална мрежа. При позитивен резултат от тази проверка се продължава към точка 27, а при негативен – точка 20;

27. Натиснат ли е IP адрес – ако след сканиране са намерени устройства в безжичната локална мрежа, потребителят може да натисне върху някой от изведените IP адреси, за да осъществи връзка с него. Тук се извършва проверка дали потребителя е натиснал върху IP адрес. При позитивен резултат от тази проверка се продължава към точка 28, а при негативен – точка 20;

28. Визуализация на графичен дизайн – изчертава се панел за свързване с посочен IP адрес, бутон за свързване, бутон за отказ и опция за избиране на типа на връзката;

29. Натиснат ли е бутон – извършва се проверка дали е натиснат бутон. При позитивен резултат се продължава към точка 30, а при негативен се извършва нова проверка;

30. Бутон за връзка - извършва се проверка дали е натиснат бутона за връзка. При позитивен резултат от тази проверка се продължава към точка 13. Негативен резултат от проверката означава, че е натиснат бутона за отказ, затова се продължава към точка 31;

31. Връщане на предходен панел - изчиства се дисплея на потребителското устройство и се изчертава панела за сканиране. След завършване на тази операция се продължава към точка 20;

3.4.4.2 Алгоритъм на работа на подпрограмата за управление.

На фиг 3.12 е представена блок диаграма на алгоритъма на работа на подпрограмата за управление.

1. Осъществяване на връзка – стартира се задача за осъществяване на връзка с избрания IP адрес. Тази задача осъществява всички протоколни механизми за създаване на канал за комуникация;

2. Визуализация на графичен дизайн - Изчиства се дисплея на потребителското устройство и се изчертава панел за управление, съдържащ бутони за увеличаване и намаляване на линейна скорост, бутони за увеличаване и намаляване на ъглова скорост, бутони за стартиране и спиране видеоприемане, бутони за управление на посоката на движение, бутон за стартиране на подпрограма за измерване и поле за визуализиране на видео;

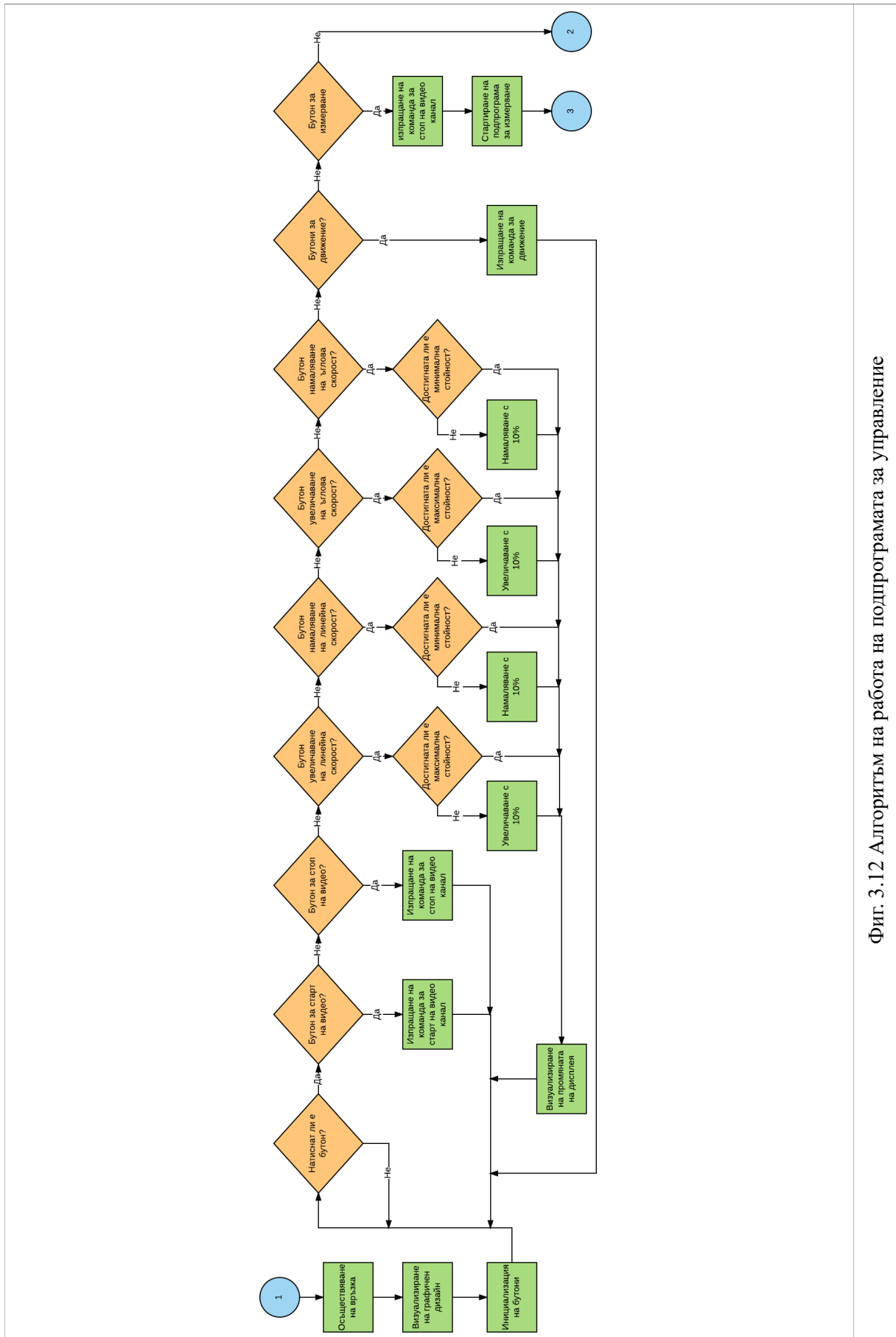
3. Инициализация на бутони - след създаването на графичното изображение на бутоните се извършва инициализация на функцията на всеки бутон;

4. Изпращане на команда за старт на видеоканал – изпраща се команда към главното управляващо устройство за стартиране на видеоканалът и изпращането на кадри, прочетени от камерата. След завършване на тази операция се продължава към точка 3 за нова проверка за натиснат бутон;

5. Бутон за старт на видео – извършва се проверка дали натиснатия бутон е за стартиране на видеоканал. При позитивен резултат се продължава към точка 6, а при негативен – точка 7;

6. Бутон за стоп на видео - изпраща се команда към главното управляващо устройство за спиране на видеоканалът и изпращането на кадри, прочетени от камерата. След завършване на тази операция се продължава към точка 4 за нова проверка за натиснат бутон;

7. Бутон за стоп на видео – извършва се проверка дали натиснатия бутон е за спиране на видеоканал. При позитивен резултат се продължава към точка 8, а при негативен – точка 9;



Фиг. 3.12 Алгоритъм на работа на подпрограмата за управление

8. Бутон за стоп на видео - изпраща се команда към главното управляващо устройство за спиране на видеоканалът и изпращането на кадри, прочетени от камерата. След завършване на тази операция се продължава към точка 4 за нова проверка за натиснат бутон;

9. Бутон за увеличение на линейна скорост – извършва се проверка дали натиснатия бутон е за увеличаване на линейна скорост. При позитивен резултат се продължава към точка 10, а при негативен – точка 12;

10. Достигната ли е максимална стойност – извършва се проверка дали е зададена максималната скорост. При позитивен резултат се продължава към точка 21, а при негативен – точка 11;

11. Увеличаване с 10% - скоростта на мобилната система за наблюдение се контролира от коефициента на запълване на широчинно-импулсните сигнали, с които се управляват моторите, затова за контролиране на скоростта се задава колко процента да бъде коефициентът на запълване на сигнала. Тук се извършва увеличение на коефициента на запълване с една стъпка от 10%. След завършване на тази операция се продължава към точка 21;

12. Бутон за намаляне на линейната скорост - извършва се проверка дали натиснатия бутон е за намаляне на линейната скорост. При позитивен резултат се продължава към точка 13, а при негативен – точка 15;

13. Достигната ли е минимална стойност – извършва се проверка дали е зададена минимална скорост. При позитивен резултат се продължава към точка 21, а при негативен – точка 14;

14. Намаляне с 10% - намалява се коефициентът на запълване с една стъпка от 10%. След завършване на тази операция се продължава към точка 21;

15. Бутон за увеличение на ъглова скорост – извършва се проверка дали натиснатия бутон е за увеличаване на ъглова скорост. При позитивен резултат се продължава към точка 16, а при негативен – точка 18;

16. Достигната ли е максимална стойност – извършва се проверка дали е зададена максималната скорост. При позитивен резултат се продължава към точка 21, а при негативен – точка 17;

17. Увеличаване с 10% - ъгловата скорост на мобилната система за наблюдение се контролира от разликата в коефициента на запълване на широчинно импулсните сигнали на левия и десния мотор, затова за контролиране на ъгловата скорост се задава колко процента да бъде разликата в коефициентите на запълване на сигналите. Тук се извършва увеличение на разликата коефициентите на запълване с една стъпка от 10%. След завършване на тази операция се продължава към точка 21;

18. Бутон за намаляне на ъглова скорост - извършва се проверка дали натиснатия бутон е за намаляне на ъгловата скорост. При позитивен резултат се продължава към точка 19, а при негативен – точка 22;

19. Достигната ли е минимална стойност – извършва се проверка дали е зададена минимална ъглова скорост. При позитивен резултат се продължава към точка 21, а при негативен – точка 20;

20. Намаляване с 10% - намалява се разликата в коефициентите на запълване с една стъпка от 10%. След завършване на тази операция се продължава към точка 21;

21. Визуализация на промяната на дисплея – визуализират се зададените стойности за на коефициента на запълване за контрол на линейна скорост и разликата в коефициентите на запълване за контрол на ъглова скорост. След завършване на тази операция се продължава към точка 4 за нова проверка за натиснат бутон;

22. Бутон за движение – потребителския интерфейс съдържа 9 бутона за управление на движение: бутон за движение напред, назад, напред и наляво, напред и надясно, назад и наляво, назад и надясно, завъртане на ляво, завъртане на дясно и стоп. Тук се извършва проверка дали е натиснат

някой от тези бутони. При позитивен резултат се продължава към точка 23, а при негативен – точка 24;

23. Изпращане на команда за задвижване – след установяване на натиск на бутон за движение се изпраща команда до главното управляващо устройство с желаната посока на движение, стойността за линейна скорост и ъглова скорост. След завършване на тази операция се продължава към точка 4 за нова проверка за натиснат бутон;

24. Бутон за измерване – извършва се проверка дали натиснатият бутон е за стартиране на подпрограмата за измерване. При позитивен резултат от тази проверка се продължава към точка 25. Негативен резултат от тази проверка означава, че натиснатият бутон е за връщане назад и програмното изпълнение се връща в подпрограмата за свързване;

25. Изпращане на команда за стоп на видеоканал - изпраща се команда към главното управляващо устройство за спиране на видеоканалът и изпращането на кадри, прочетени от камерата. След завършване на тази операция се продължава към точка 26;

26. Стартиране на подпрограмата за измерване – подготвя се дисплея за изчертаване на панел за измерване и се стартира подпрограмата за измерване.

3.4.4.3 Алгоритъм на работа на подпрограмата за измерване

На фиг 3.13 е представена блок диаграма на алгоритъма на работа на подпрограмата за управление.

1. Конфигуриране на дисплей – изчертава се потребителския интерфейс за визуализиране на измервания от сензорите, съдържащ седем полета за извеждане на информация от сензорите и бутон за връщане назад;

2. Натиснат ли е бутон за връщане назад – извършва се проверка дали е натиснат бутон за връщане назад. При позитивен резултат от тази проверка се подготвя дисплея за изчертаване на панела за управление и програмното

изпълнение се връща в подпрограмата за управление. При негативен резултат от проверката се продължава към точка 3;

3. Време за прочитане – прочитане на информация за температура, влажност и дистанция и опресняването на информацията на дисплея се извършва през предварително зададен интервал от време. Тук се проверява дали този интервал е изтекъл. При позитивен резултат от проверката се продължава към точка 4, а при негативен - точка 2;

4. Изпращане на команда за прочитане на температура и влажност – изпраща се команда до главното управляващо устройство за прочитане на информация за последно измерените стойности на температура и относителна влажност;

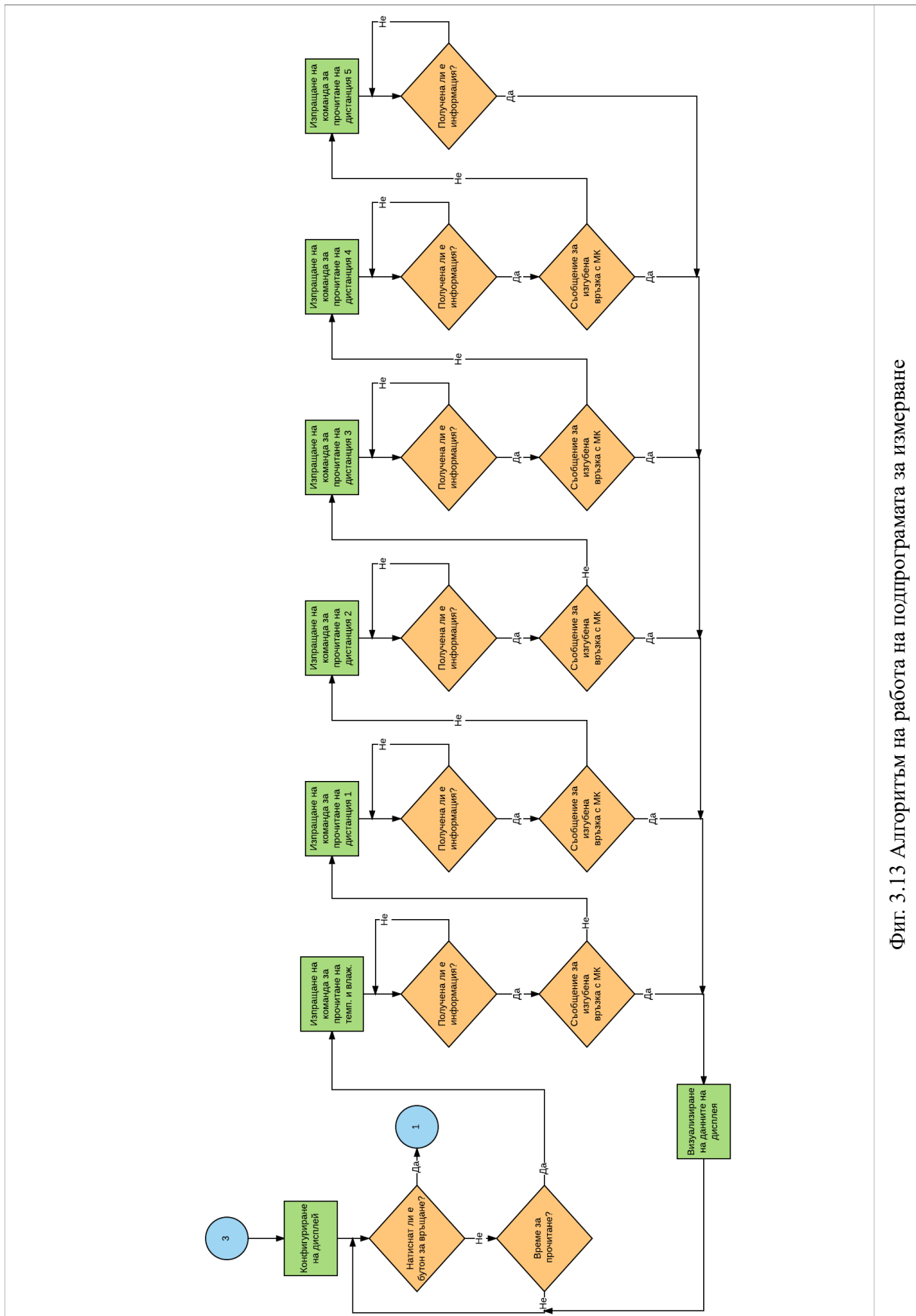
5. Получена ли е информация – извършва се проверка за отговор на главното управляващо устройство. При позитивен резултат от тази проверка се продължава към точка 6, а при негативен се извършва нова проверка;

6. Съобщение за изгубена връзка с микроконтролер – проверява се дали полученото съобщение от главното управляващо устройство е за изгубена връзка с микроконтролера. При позитивен резултат от тази проверка се продължава към точка 21, а при негативен – точка 7;

7. Изпращане на команда за прочитане на дистанция 1 – изпраща се команда до главното управляващо устройство за прочитане на информация за последно измерената стойност от първия сензор за дистанция;

8. Получена ли е информация – извършва се проверка за отговор на главното управляващо устройство. При позитивен резултат от тази проверка се продължава към точка 9, а при негативен се извършва нова проверка;

9. Съобщение за изгубена връзка с микроконтролер – проверява се дали полученото съобщение от главното управляващо устройство е за изгубена връзка с микроконтролера.



Фиг. 3.13 Алгоритъм на работа на подпрограмата за измерване

При позитивен резултат от тази проверка се продължава към точка 21, а при негативен – точка 10;

10. Изпращане на команда за прочитане на дистанция 2 – изпраща се команда до главното управляващо устройство за прочитане на информация за последно измерената стойност от втория сензор за дистанция;

11. Получена ли е информация – извършва се проверка за отговор на главното управляващо устройство. При позитивен резултат от тази проверка се продължава към точка 12, а при негативен – точка 11;

12. Съобщение за изгубена връзка с микроконтролер – проверява се дали полученото съобщение от главното управляващо устройство е за изгубена връзка с микроконтролера. При позитивен резултат от тази проверка се продължава към точка 21, а при негативен – точка 13;

13. Изпращане на команда за прочитане на дистанция 3 – изпраща се команда до главното управляващо устройство за прочитане на информация за последно измерената стойност от третия сензор за дистанция;

14. Получена ли е информация – извършва се проверка за отговор на главното управляващо устройство. При позитивен резултат от тази проверка се продължава към точка 15, а при негативен се извършва нова проверка;

15. Съобщение за изгубена връзка с микроконтролер – проверява се дали полученото съобщение от главното управляващо устройство е за изгубена връзка с микроконтролера. При позитивен резултат от тази проверка се продължава към точка 21, а при негативен – точка 16;

16. Изпращане на команда за прочитане на дистанция 4 – изпраща се команда до главното управляващо устройство за прочитане на информация за последно измерената стойност от четвъртия сензор за дистанция;

17. Получена ли е информация – извършва се проверка за отговор на главното управляващо устройство. При позитивен резултат от тази

проверка се продължава към точка 18, а при негативен се извършва нова проверка.

18. Съобщение за изгубена връзка с микроконтролер – проверява се дали полученото съобщение от главното управляващо устройство е за изгубена връзка с микроконтролера. При позитивен резултат от тази проверка се продължава към точка 21, а при негативен – точка 19;

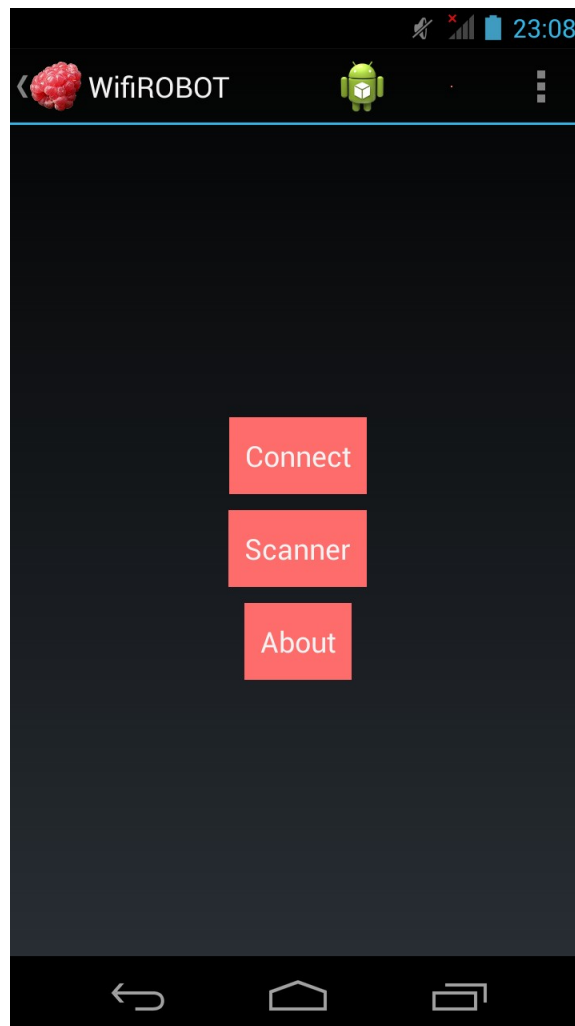
19. Изпращане на команда за прочитане на дистанция 4 – изпраща се команда до главното управляващо устройство за прочитане на информация за последно измерената стойност от четвъртия сензор за дистанция;

20. Получена ли е информация – извършва се проверка за отговор на главното управляващо устройство. При позитивен резултат от тази проверка се продължава към точка 21, а при негативен се извършва нова проверка;

21. Визуализиране на данните на дисплея – извежда се информация за състоянието на връзката с микроконтролера и стойностите за температурата, относителната влажност и дистанция. След изпълнение на тази операция се продължава към точка 2.

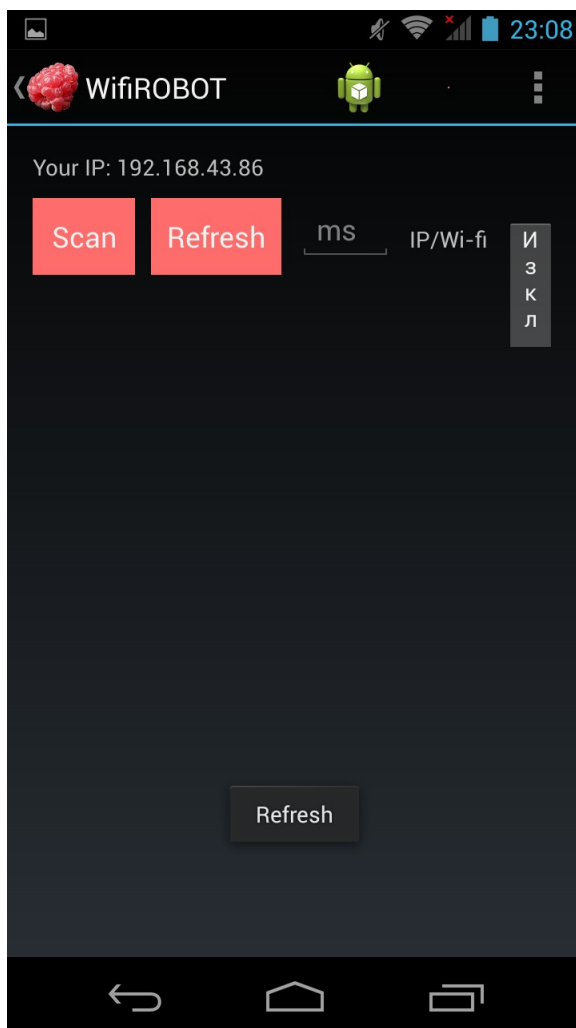
**Глава 4. УКАЗАНИЯ ЗА НАЧИН НА РАБОТА С
МОБИЛНАТА СИСТЕМА ЗА НАБЛЮДЕНИЕ**

Контролът на мобилната система за наблюдение, както бе упоменато по – рано, се извършва от мобилен телефон, работещ с операционна система „Android”. Максимум до минута след стартиране на системата, тя преминава в готовност за работа и потребителят може да установи връзка с нея. След стартиране на приложението, на дисплея на мобилния телефон се визуализира начален екран, който е представен на фиг. 4.1.

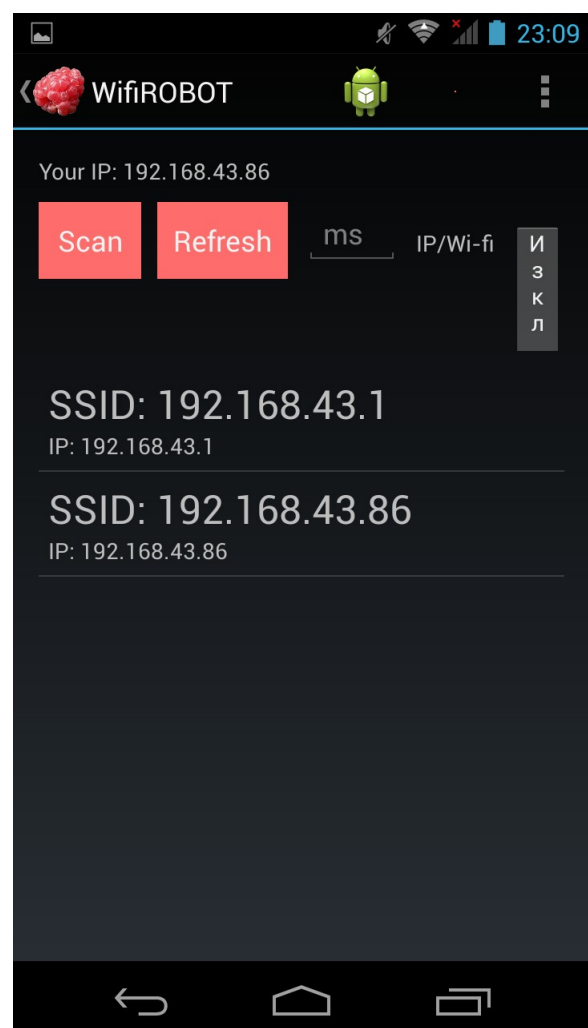


Фиг. 4.1 Начален екран на потребителската програма

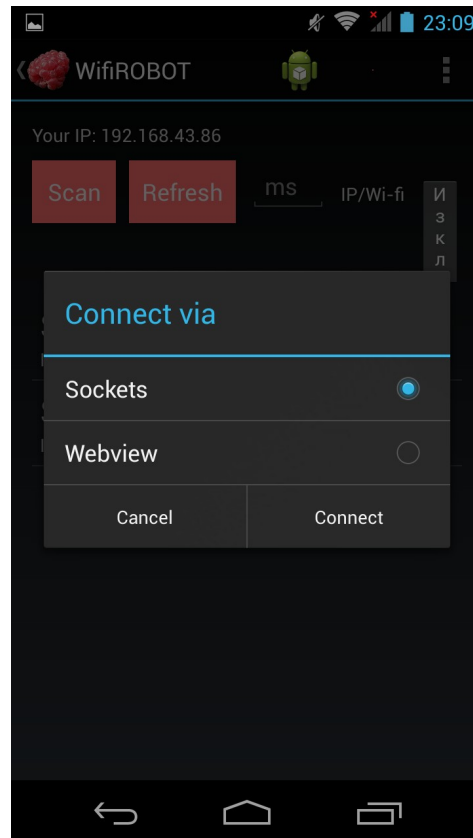
При натискане на бутона „Scanner” се преминава към екран за сканиране, визуализиран на фиг. (4.2.а). В горната лява част на екрана се изписва IP адреса на потребителското устройство или съобщение „Not connected” ако няма връзка с безжична виртуална мрежа. След установяване на връзка с мрежа, може да се направи сканиране за налични устройства в мрежата, чрез натискане на бутона „Scan”. Представено е изображение на резултат от сканиране с намерени два IP адреса на фиг. (4.2.б) За да осъществи опит за връзка с някое от устройствата в мрежата, потребителят може да натисне върху IP адреса на устройството. След това потребителя трябва да избере типа на връзката, чрез натиск на една от двете опции визуализирани на фиг. 4.3.



Фиг.4.2.а Екран за сканиране



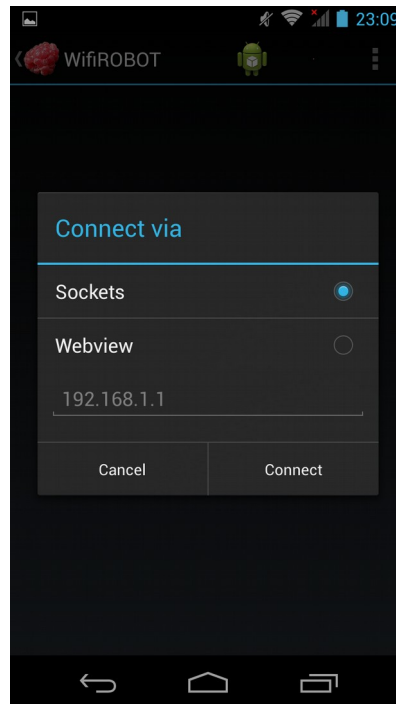
Фиг. 4.2.б Резултат от сканиране



Фиг. 4.3 Опции за типа на връзка

За директно осъществяване на връзка с мобилната система за наблюдение, чрез въвеждане на IP адрес, потребителят може да натисне бутон „Connect”, докато се намира на началния екран, показан на фиг 4.1. След натискане на този бутон на дисплея се визуализира панел за директна връзка чрез въвеждане, показан на фиг .4.4. Потребителят трябва да въведе IP адреса на мобилната система за наблюдение в полето за въвеждане, след което да избере метод за връзка „Socket” и да натисне бутон „Connect”. При желание за връщане в началния панел, потребителя може да откаже връзката чрез натискане на бутон „Cancel”.

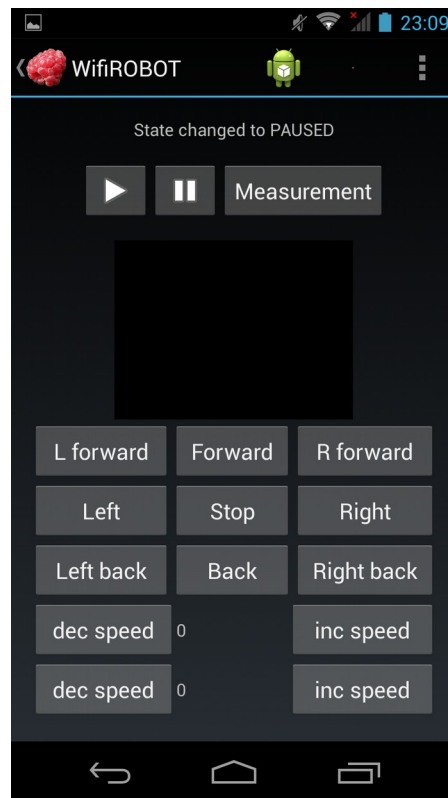
След осъществяване на връзка с мобилната система за наблюдение на дисплея на мобилния телефон се визуализира панел за управление, показан на фиг. 4.5.



Фиг. 4.4 Панел за връзка чрез въвеждане на IP адрес

Панелът съдържа следните елементи:

- Бутон за старт на видео – стартира видео предаването;
- Бутон за стоп на видео – спира видео предаването;
- Панел за видео – визуализиране на видео изображението;
- Бутон за увеличаване на линейна скорост – увеличаване на коефициентът на запълване на сигнала управляващ моторите, с 10 %;
- Бутон за намаляване на линейна скорост – намаляване на коефициентът на запълване на сигнала управляващ моторите, с 10 %;
- Бутон за увеличаване на ъглова скорост – увеличаване на разликата между коефициентите на запълване на сигнала управляващ вътрешния, за конкретния завой мотор спрямо сигнала управляващ външния мотор с 10%;
- Бутон за намаляване на ъглова скорост – намаляване на разликата между коефициентите на запълване на сигнала, управляващ вътрешния за конкретния завой мотор, спрямо сигнала управляващ външния мотор с 10%;

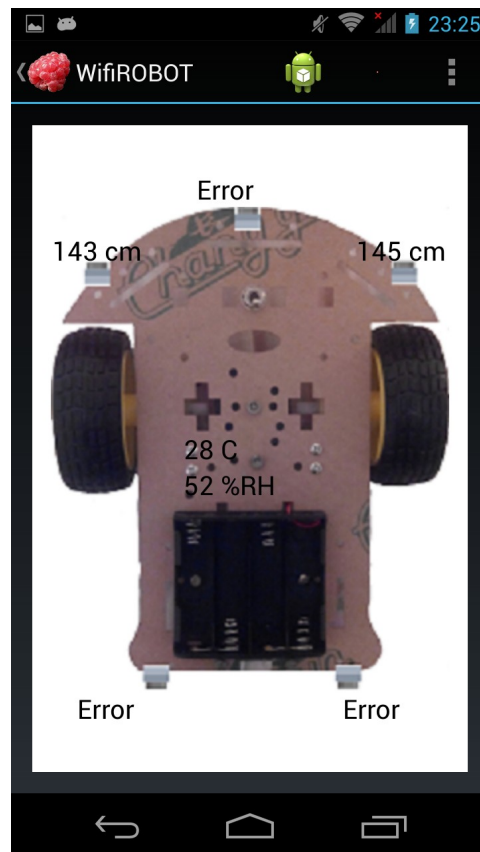


Фиг. 4.5 Панел за управление на мобилната система за наблюдение

- Бутон за движение напред – задвижване на мобилната система напред със зададената линейна скорост;
- Бутон за движение назад – задвижване на мобилната система назад със зададената линейна скорост;
- Бутон за завъртане наляво – завъртане на мобилната система наляво със зададената ъглова скорост (не се извършва линейно движение);
- Бутон за завъртане надясно – завъртане на мобилната система надясно с зададената ъглова скорост (не се извършва линейно движение);
- Бутон за движение напред и наляво – задвижване на мобилната система напред със зададената линейна скорост и наляво със зададената ъглова скорост;
- Бутон за движение напред и надясно – задвижване на мобилната система напред със зададената линейна скорост и надясно със зададената ъглова скорост;

- Бутон за движение назад и наляво – задвижване на мобилната система назад със зададената линейна скорост и наляво със зададената ъглова скорост;
- Бутон за движение назад и надясно – задвижване на мобилната система назад със зададената линейна скорост и надясно със зададената ъглова скорост;
- Бутон за стартиране на измерване – визуализиране на панел за измерване показан на фиг 4.6.

Панелът за измерване разполага със 7 полета за извеждане на данни. Температурата и относителната влажност се визуализират в центъра на дисплея, а информацията за дистанция - до съответното изображение на сензор за дистанция, за който се отнасят данните в това поле. При наличие на грешка в някой от сензорите на екрана се извежда съобщение „Error” в полето отговарящо за съответния сензор



Фиг. 4.6 Панел за визуализиране на измервания

ЗАКЛЮЧЕНИЕ И НАСОКИ НА РАЗВИТИЕ НА СИСТЕМАТА

Разработената мобилна система за наблюдение позволява извършването на наблюдение в реално време от мобилен телефон. Системата разполага с динамичен контрол на скоростта на линейно движение и скоростта на ъглово движение, което позволява прецизен контрол. Реализирана е и възможност за измерването на дистанция до обекти и на температура и относителна влажност на околната среда. За развитие на системата е предвидено в бъдеще добавяне на следните възможности: автономно управление и изчертаване на карта на затворени помещения, прецизно локализиране в затворени помещения, разпознаване на обекти чрез анализ на изображението от камерата, гласово управление.

ЛИТЕРАТУРА

1. Атанасов, А., (2007), Основи на микропроцесорната техника, София, 2007
2. Атанасов, А., (2007), Основи на импулсната и цифровата схемотехника, София
3. Овчаров, С., Автоматизация на електронното производство
4. Fridrick M. Cady, (1997), Microcontrollers and Microcomputers, Oxford University Press
5. http://www.keil.com/dd/docs/datashts/arm/cortex_m0/r0p0/dui0497a_cortex_m0_r0p0_generic_ug.pdf
6. http://www.microsemi.com/document-portal/doc_view/130725-cortex-m1-technical-reference-manual
7. http://www.keil.com/dd/docs/datashts/arm/cortex_m3/r2p0/ddi0337g_cortex_m3_r2p0_trm.pdf
8. http://www.keil.com/dd/docs/datashts/arm/cortex_m4/r0p1/dui0553a_cortex_m4_dgug.pdf
9. http://infocenter.arm.com/help/topic/com.arm.doc.dui0646a/DUI0646A_cortex_m7_dgug.pdf
10. <https://tools.ietf.org/html/rfc768>
11. <https://tools.ietf.org/html/rfc2460>
12. <https://tools.ietf.org/html/rfc675>
13. <https://tools.ietf.org/html/rfc793>
14. http://www.nxp.com/documents/user_manual/UM10204.pdf
15. <http://www.totalphase.com/support/articles/200349176>
16. <http://mipi.org/specifications/camera-interface>