

Додаток 2: Подвиртуелни инструменти (ПВИ)

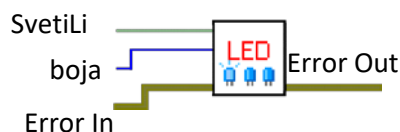
Со цел програмата да ги има сите карактеристики набројани во текстот, изработени беа повеќе ПВИ кои извршуваат посебни функции.

ПВИ-то „razlika.vi“ (слика 2.1) на влез добива две променливи од тип unsigned long и ја пресметува разликата помеѓу нив. Се користи за пресметка на период во ms помеѓу два момента.

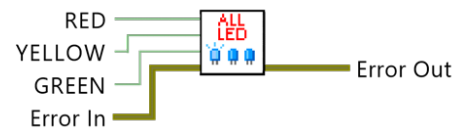
ПВИ-то „LED.vi“ (слика 2.2) се користи за управување на LED, со тоа што на влез добива BooleanSvetiLi за тоа дали LED-от со дадена „боја“ треба да свети или не. Во случај да има промена на состојбата на LED, се повикува ПВИ „Digital Write“ за запишување на дигитален сигнал што е дел од LINUX-пакетот. Ова се прави со цел да се заштеди времето потребно за извршување на дигитален запис, а бидејќи Ардуиното ја држи состојбата на пиновите на нивната последна зададена вредност – истото нема да влијае врз функционалноста на програмата.



2.1 ПВИ „razlika.vi“



2.2 ПВИ „LED.vi“



2.3 ПВИ „allLED.vi“

ПВИ-то „allLED.vi“ во себе содржи три ПВИ „LED.vi“ за секоја LED посебно, па на влез добива само три Boolean вредности за трите LED по редослед прикажан на слика 2.3.

ПВИ-то digChit.vi (слика 2.4) врши отчитување на состојбата на зададениот дигитален пин.

ПВИ chitKop.vi (слика 2.5) врши отчитување на вредноста на физичкото и виртуелното копче, враќа информација за тековната состојба на копчето, но и за тоа дали истото комутирало или не.

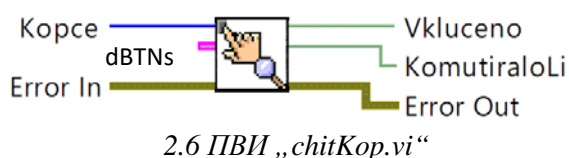
ПВИ-то sirena.vi (слика 2.6) на влез добива Boolean порменлива Sviri согласно која ја вклучува или исклучува сирената. Исто како и кај LED.vi, ПВИ-то врши запишување на дигитален сигнал само во случај да тековната состојба на сирената е различна од онаа што се задава.



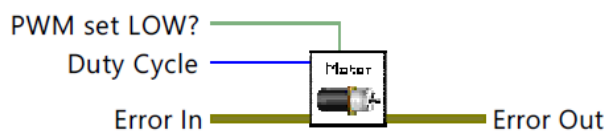
2.4 ПВИ „digChit.vi“



2.5 ПВИ „sirena.vi“



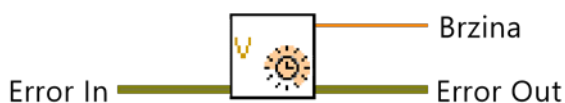
2.6 ПВИ „chitKop.vi“



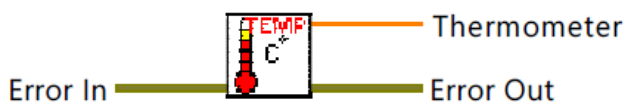
2.7 ПВИ „motor.vi“

ПВИ-то motor.vi (слика 2.7) врши управување со факторот на исполнетост (Duty Cycle) на пинот на моторот со тоа што го нагудува на вредност зададена на неговиот влез. Од исти причини како претходно, „аналогното“ запишување се извршува доколку има промени од претходната негова вредност. Вредноста на факторот на исполнетост треба да биде нагодена како контролата на пинот на моторот да е Active-HIGH. Но, во случај електронското коло (види поглавје “Електрично коло за контрола на DC-мотор“) е направено така што пинот за контрола на моторот треба да биде поставен на Active-LOW како што е тука, на влез за „PWM set LOW?“ треба да се проследи вредност „true“, така што ПВИ ќе ја нагоди вредноста соодветно. ПВИ-то исто така води евиденција за тековната состојба на моторот, како и времето на негова последна комутација.

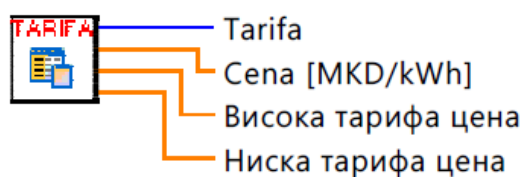
ПВИ-то brzinoMetar.vi (слика 2.8) врши следење на состојбата на индуктивното сетило и го мери времето помеѓу две комутации. Потоа користејќи ја пресметува тековната брзина на вртење на начин објаснет во поглавјето “Индуктивно сетило и принцип на мерење на брзината“. Се смета дека доколку не се случи комутација во период подолг од 6 секунди, што соодветствува на брзина од 5 врт/мин, тогаш системот е запрен и за тековна брзина се зема 0. Се пресметува аритметичка средина на последните 6 вака добиени вредности и таквиот податок се проследува на излез од ПВИ-то.



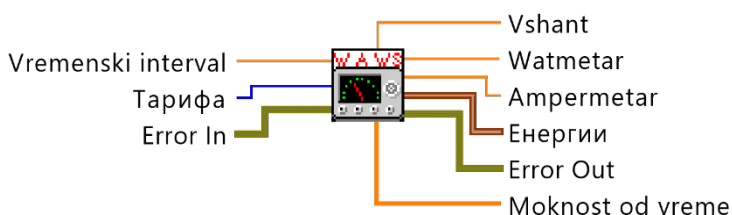
2.8 ПВИ „brzinoMetar.vi“



2.9 ПВИ „Termometer.vi“



2.10 ПВИ „Tarifa.vi“



2.11 ПВИ „Multimer.vi“

ПВИ-то Termometer.vi (слика 2.9) врши мерење на напонот на аналогниот влез одреден за колото на температурното сетило, па согласно релациите изведени во поглавјето “Електрично коло за температурното сетило“ се пресметува тековната температура. Мерењето се врши на секои 5 минути.

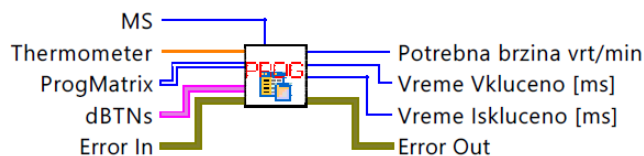
ПВИ-то Tarifa.vi (слика 2.10) според времето и денот во седмицата одредува по која тарифа се наплатува електричната енергија (ВТ – висока тарифа 5,56 МКД/kWh, НТ – ниска тарифа 2,78 МКД/kWh), па сите овие параметри ги проследува на излез.

ПВИ-то Multimer.vi (слика 2.11) на влез добива информација за тоа во која тарифа е енергијата што тековно се троши. Влезот „Vremenski interval“ одредува на колку секунди да се извршува мерењето на јачината на електрична енергија и во случај да не се поврзе никаква вредност на овој терминал, ПВИ зема вредност 5 секунди. ПВИ врши мерење на

напонот на шантот и, како што е објаснето во поглавјето “Електрично коло за мерење на јачината на електричната струја“, пресметува јачина на електрична струја, тековна потрошувачка во W, енергија потрошена во интервалот од 5 секунди, вкупно потрошена енергија, енергија потрошена само при ниска тарифа и енергија потрошена при висока тарифа. Исто така формира матрица на потрошувачка и време во секунди кога е измерена таа потрошувачка. Сите овие параметри ги проследува на неговиот излез.



2.12 ПВИ „MS.vi“



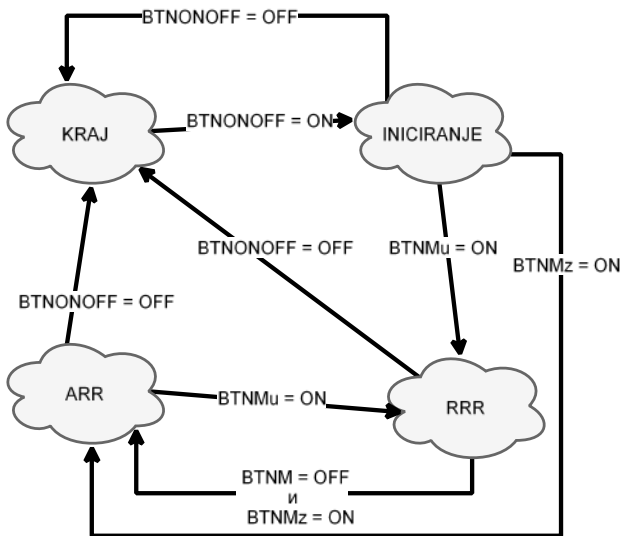
2.13 ПВИ „Programator.vi“

ПВИ-то MS.vi (слика 2.12) има единствена цел да управува со машината на состојби на системот. Предвидени се два режими на работа: автоматски режим на работа (APP) и рачен режим на работа (PPP) коишто се разликуваат по тоа кој е оној што ја врши контролата на вклучување на моторот, како и неговата брзина за што ќе стане збор подоцна. Двава режима на работа заедно со состојба за започнување и состојба за завршување на програмата ја формираат машината на состојби. Истите се именувани соодветно со „INICIRANJE“, „ARR“, „RRR“ и „KRAJ“.

MS.vi на влез добива кластер со вредности на виртуелните копчиња, како и податок за тековната температура. За време на неговото извршување, истиот врши отчитување на состојбите на физичките копчиња, па имајќи ги предвид отчитаните вредности, како и податоците добиени на влез, одлучува во која состојба да се премине во наредната итерација на главниот круг. Начинот на кој се носи одлука за промена на состојбакористејќи ги копчињата е прикажан на слика 2.14. Исто така, во случај системот да е во PPP, а моторот е исклучен подолго од 5 минути за температури над 50 °C, односно 10 минути за температури под 50 °C, MS.vi ќе ја смени состојбата во APP без разлика што ниту едно копче не е притиснато.

ПВИ-то Programator.vi (слика 2.13) на влез добива повеќе параметри според кои носи одлука за вредноста на потребната брзина на вртење на моторот кон која системот треба да се прилагоди, за колку време моторот треба да биде вклучен, а колку исклучен – наизменично во ms. Programator.vi ова го прави преку отчитување на физичките и виртуелните копчиња или преку отчитување на потребната вредност од програматорската матрица ProgMatrix зависно од тоа во кој режим на работа е системот. Ако системот се наоѓа во PPP и копчето за вклучување на моторот, VTNM, е вклучено, тогаш со секое притискање на копчето за забрзување, VTNMz, потребната брзина се инкрементира за 5 врт/мин сè до максимална вредност од 8 врт/мин, додека со притискање на копчето за успорување, VTNMu, потребната брзина ќе се декрементира за 5 врт/мин сè до вредност 0. Вредностите за „Vreme iskluceno“ и „Vreme vkluceno“ се отчитуваат од програматорската матрица, но сепак не играат никаква улога во работата на системот во PPP. Вредноста за потребната брзина не се менува со притискање на копчињата за забрзување и успорување во случај копчето за вклучување на моторот да е исклучено, тогаш истите служат исклучиво за менување на режимот на работа како што е објаснето на сликата 2.14. Во APP, вредноста за потребната брзина се отчитува од програматорската матрица, за која ако не е зададено преку контролната табла, се зема дифолт матрицата прикажана на слика 2.15. Во првата редица на матрицата, се вредности на горните граници на интервалот на температура, па programator.vi прво проверува во која колона е интервалот во кој се наоѓа тековната температура. Потоа, за потребна брзина ја зема вредноста во втората редица на истата

колона. Во третата редица е зададена процентуална вредност од интервал Тт во кој моторот треба да биде вклучен, па така времињата за вклученост/исклученост се пресметуваат така што бројот од третата редица на истата колона се множи со 0,01, па со временскиот интервал Тт што изнесува 5 минути претворени во ms за температури над 50°C, а за температури под 50°C – 10 минути.

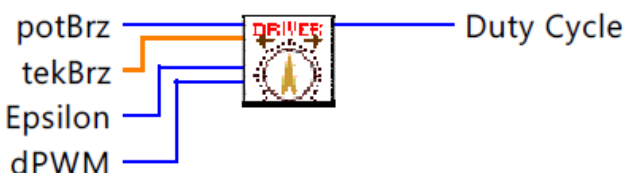


2.14 Управување со машината на состојби

30	50	70	200
10	10	25	30
10	30	60	95

2.15 Дифолт програматорска матрица

ПВИ-to driverM.vi (слика 2.16) на влез добива податоци за потребната брзина на вртење на моторот во врт/мин, тековната брзина на вртење, како и незадолжителни параметри Epsilon и dPWM кои по дифолт изнесуваат 10 и 5 соодветно. Целта на овој ПВИ е да ја нагоди вредноста на факторот на исполнетост таква што тековната брзина на вртење на моторот биде во интервалот $[potBrz * (1 - Epsilon / 100), potBrz * (1 + Epsilon / 100)]$. Во случај тековната брзина да биде вон овој интервал, ПВИ врши корекција на факторот на исполнетост за вредност од dPWM во позитивна или негативна насока зависно потребата. Кога вредноста ќе биде коригирана, ПВИ нема да изврши нова корекција сè дури не измине време пресметано според релацијата $t_{min} = \Delta\theta / \omega_{tek}$. Во случај на вклучување на моторот се зема почетна вредност за факторот на исполнетост на 13%, а при исклучување веднаш се зема вредност 0. Ваквата корекција се прави така што се претпоставува дека моторот е контролиран од Active-HIGH сигнал и како таква вредноста се проследува понатаму. ПВИ motor.vi, како што е објаснето погоре, врши нагудување на вредноста земајќи предвид со каков сигнал е контролиран моторот. Вредностите за Epsilon и dPWM можат да се нагодат и од контролната табла.



2.16 ПВИ „driverM.vi“



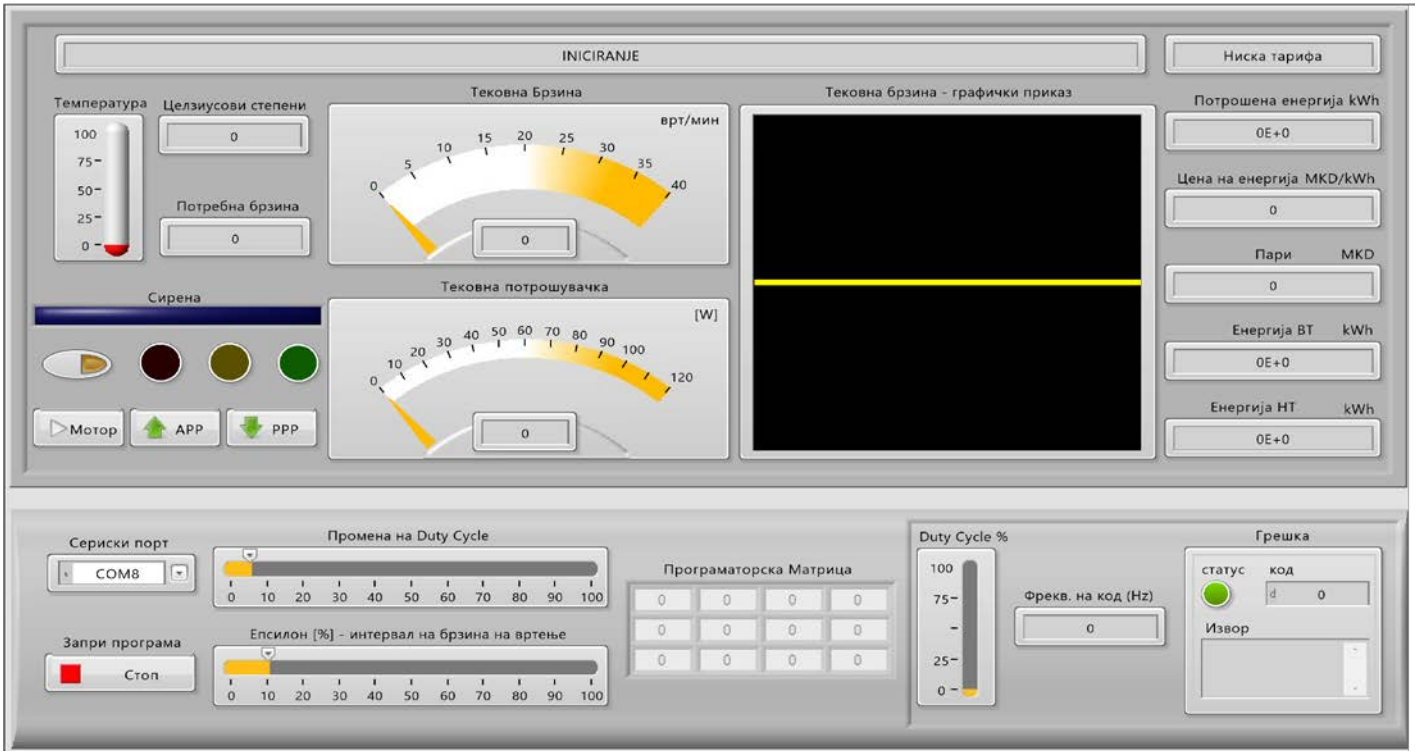
2.17 ПВИ „GrafSimBrz.vi“

ПВИ GrafSimBrz.vi (слика 2.17) на влез добива информација само за тековната брзина на вртење во врт/мин, па врши симулација и графички приказ на физичкото вртење на мешалката. Истото е изведено користејќи XY графа со вредности за X од $-1 * 10^{-5}$ и $1 * 10^{-5}$, а вредностите на Y пресметани според формулата

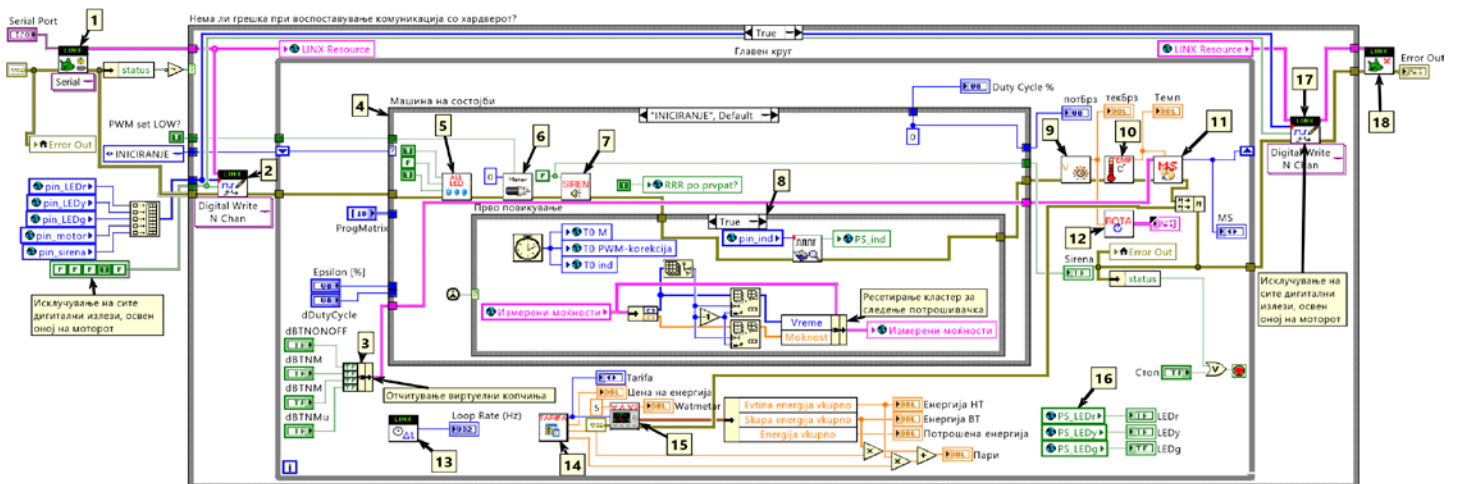
$$y_i = x * \tan(\theta_{i-1} + \omega * \Delta t), \quad \theta_i = \theta_{i-1} + \omega * \Delta t$$

Каде што: θ_{i-1} - агол во рад што го зафаќа правата у во претходна итерација,
 ω - тековна брзина во rad/s,
 Δt - време изминато од последното повикување на ПВИ во секунди.

Вака добиените вредности по пребришување на графикот се цртаат на графа на секое извршување на ПВИ-то. На овој начин успешно се симулира вртењето на мешалката во реално време.

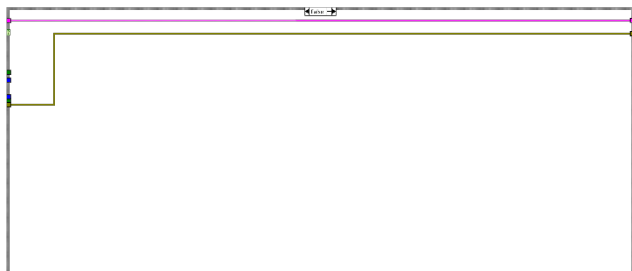


2.18 Контролна табла на програмата во LabVIEW



2.19 Програмски дијаграм во LabVIEW

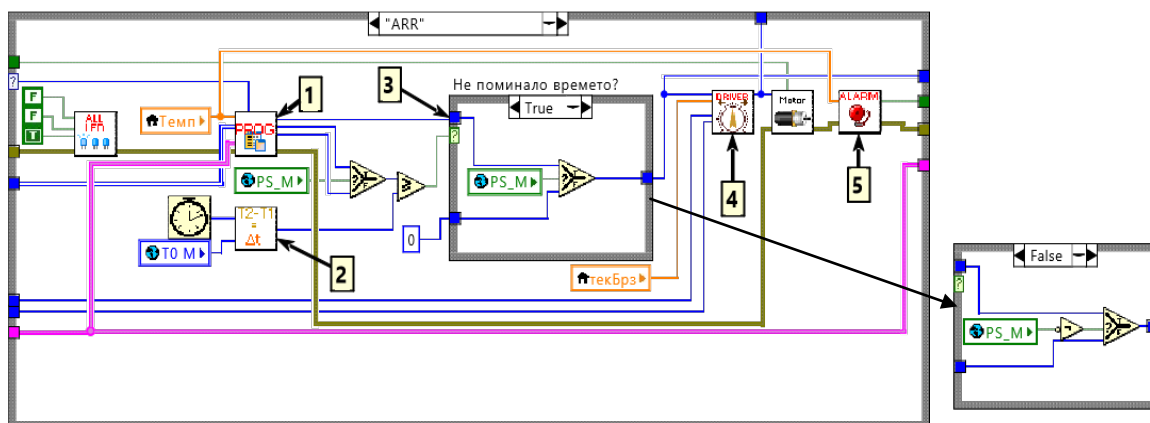
Дијаграм на главниот код



2.20 Грешка при воспоставување комуникација со Arduino-то - Дијаграм

Контроланата табла (КТ) на програмата во LabView е дадена на слика 2.18, а дијаграмот на главниот код е прикажан на слика 2.19. Со самото стартување на програмата се воспоставува комуникација помеѓу компјутерот и Arduino-то преку отворање на сериска комуникација со ПВИ означен со (1) на слика 2.19. На влез ПВИ (1) добива празен кластер на грешка и информација на кој сериски порт е приклучено Arduino-то. Во

случај на излезот од ПВИ (1) да се добие информација дека настанала грешка (true), статусот се инвертира во false и се извршува кодот на истоимениот случај на Case-структурата (слика 2.20). Се продолжува до ПВИ (18) кој ја затвора комуникацијата и со тоа завршува програмата. Најчеста причина за појава на грешка тука е физичката неповрзаност на Arduino-то со компјутерот, поставување погрешен сериски порт на влез на ПВИ (1) или неинсталирани драјвери.



2.21 Дијаграм на код за APP

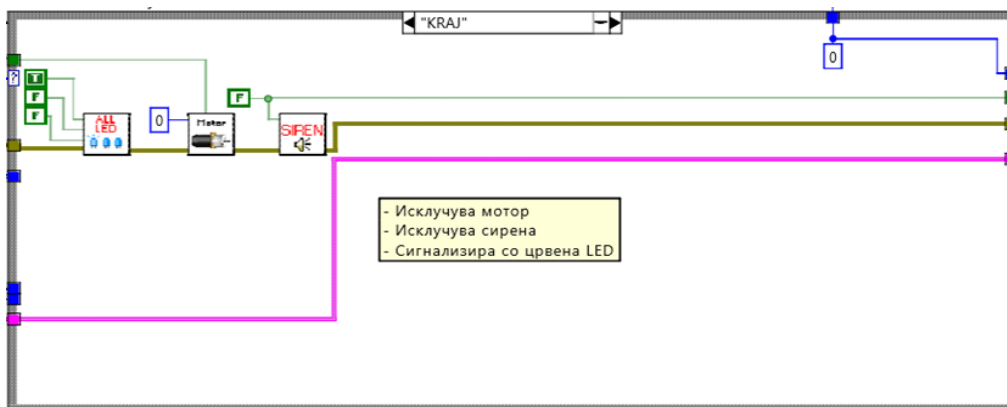
Откако успешно ќе се воспостави комуникацијата, се извршува кодот на случајот „true“ од Case-структурата како што е и прикажано на слика 2.19. Пред да се влезе во „главниот круг“ сите дигитални излези со ПВИ (2) се поставуваат на исклучено, освен дигиталниот излез со кој се контролира електричниот мотор – тој се поставува на вклучено со цел моторот да биде исклучен (види „Електрично коло за контрола на DC-мотор“).

Со влегувањето во прва итерација на главниот круг програмата е успешно стартувана. До структурата „Машина на состојби“ се носи кластер вредности отчитани од виртуелните копчиња на контролната табла (слика 2.19 (3)), како и вредност за состојба од претходна итерација (што за нулта итерација е INICIRANJE).

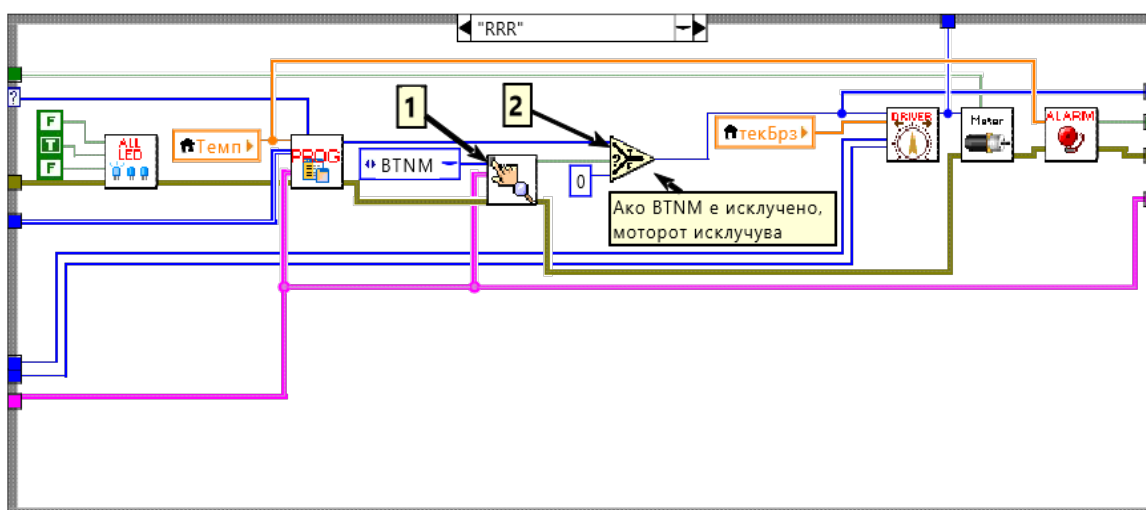
Ако системот се наоѓа во состојбата „INICIRANJE“ тогаш моторот и сирената се исклучуваат, додека црвената и зелената LED се вклучени. Ако по првпат системот се наоѓа во оваа состојба, се извршува кодот во структурата „прво извршување“ каде што се ажурираат вредностите за глобалните променливи за време, се проверува состојбата на индуктивното сетило и се чисти матрицата на вредностите на времето и измерената потрошувачка од претходното извршување на кодот. Понатаму, овој дел воопшто не се извршува сè додека кодот не биде рестартиран.

Ако системот се наоѓа во состојбата „ARR“ тогаш се извршува кодот од Case-структурата прикажан на слика 2.21.

На влез се сигнализира со запалување на зелена LED дека системот се наоѓа во APP, па потоа се извршува ПВИ programator.vi (слика 2.21 (1)) кој согласно тековната температура одлучува за вредноста на потребната брзина, како и за времето на вклученост/исклученост на моторот. До операторот за споредување поголемо или еднакво, се носи вредноста на времето на вклученост или исклученост зависно тековната состојба на моторот и вредност за изминато време од последна комутација на моторот (слика 2.21 (2)). Ако времето на вклученост/исклученост е поголемо од изминатото време, моторот останува во таа состојба во која е и тоа е изведено со проследување на соодветната вредност за потребната брзина во состојбата „true“ на структурата „Не поминало времето?“ (слика 2.21 (3)). Во спротивно, моторот ќе треба да ја смени својата состојба, па извршувајќи го кодот од состојбата „false“ на истата структура, се проследува вредност за потребна брзина предодредена за спротивната состојба на тековната на моторот. Понатаму, при извршување на ПВИ motor.vi оваа промена ќе биде забележана и глобалните променливи за тековната состојба на моторот, заедно со времето на последна комутација на истиот ќе бидат ажурирани, така што во наредна итерација нема да настане повторна промена на состојбата сè додека потребното време не е поминато. Вредноста за потребната брзина се проследува до ПВИ driverM.vi(слика 2.21 (4)) и тој истата ја преведува во вредност за факторот на исполнетости ја проследува до ПВИ motor.vi. На крајот од состојбата APP на структурата „Машина на состојби“ се извршува ПВИ alarm.vi (слика 2.21 (5)) што алармира доколку температурата надмине одредена граница или почнува да паѓа.



2.22 Дијаграм на код за KPAJ



2.23 Дијаграм на код за PPP

Ако системот се наоѓа во состојба „RRR“ се извршува кодот прикажан на дијаграмот на сликата 2.22. Овој код се разликува од кодот за „ARR“ само по начинот на кој се одлучува дали моторот да се вклучи или не. Истото се извршува со ПВИ chitKop.vi што ја отчитува состојбата на копчето за вклучување на моторот, BTNM, и во случај да е вклучено, до driverM.vi се проследува вредноста на потребната брзина на вртење од programator.vi, а доколку е исклучено – се проследува вредност 0. Сигнализацијата за PPP се прави со вклучување на жолтата LED.

Ако системот се наоѓа во состојба „KRAJ“, тогаш се извршува кодот прикажан на дијаграмот на сликата 2.23. Сигнализацијата се прави со вклучување на црвена LED додека моторот и сирената се исклучуваат.

Паралелно со структурата „Машина на состојби“ се извршуваат ПВИ tarifa.vi и multimer.vi означени со (14) и (15) на сликата 2.19. Tarifa.vi проследува вредности за тековната тарифа до multimer.vi, но истото го прикажува и на соодветниот индикатор на КТ што се наоѓа во горниот десен агол прикажано на сликата 2.18. Освен овој податок до КТ се проследуваат податоците за тековната цена на чинење на електричната енергија, како и цената на чинење на енергијата при ВТ и НТ за споредба. Откако ќе се изврши ПВИ tarifa.vi, се извршува ПВИ multimer.vi кој ја мери потрошувачката на моќност и пресметува количество потрошена енергија категоризирана според тарифата, како и сума на вкупно потрошената енергија до моментот. Податокот за тековната потрошувачка се проследува до индикатор на КТ, додека кластерот што ги содржи информациите за енергијата се деструктурира, вредноста за вкупно потрошената енергија до дадениот момент се проследува до соодветниот индикатор на КТ, додека вредностите за количината потрошена енергија при ВТ и НТ се множат со цената на ЕЕ во таа тарифа и се собираат со што се добива вкупната цена што треба да се плати за потрошената ЕЕ до тој момент. Истото се проследува до индикаторот „Пари“ на КТ.

По извршувањето на кодот од структурата „Машина на состојби“ се врши мерење на тековната брзина на вртење на мешалките со ПВИ brzinoMetar.vi (слика 2.19 (9)) и истото се проследува до индикаторот „Тековна брзина“ на КТ. Вредноста за тековната брзина исто така се проследува до ПВИ GrafSimBrz.vi (слика 2.19 (12)), кој врши симулација на вртењето и истото се прикажува на графикот „Тековна брзина – графички приказ“ на КТ. Ова е изработено со цел операторот на машината да има физичка претстава за тоа колку навистина брзо се врти мешалката. На индикаторот „потребна брзина“ се проследува информација за брзината која системот се обидува да ја постигне.

По мерењето на тековната брзина се врши мерење на тековната температура со ПВИ termometar.vi (слика 2.19 (10)) на секои 5 минути и истата информација се пренесува на КТ до индикаторот „температура целзиусови степени“ и е прикажана графички на термометар и бројчано. Потоа со ПВИ MS.vi (слика 2.19 (11)) се одлучува во која состојба ќе биде системот во наредната итерација и истото се проследува до КТ на индикаторот најгоре.

Во делот (16) на дијаграмот на сликата 2.19, се врши ажурирање на состојбите на виртуелните ЛЕД што се наоѓаат на КТ над копчињата за управување. Од излезот на структурата „Машина на состојби“ се добива информација за тековната состојба на сирената и истото се проследува до индикаторот „сирена“ на КТ, па кога сирената ќе биде вклучена, ќе се симулира со палење на LED-индикаторот на КТ.

На КТ во редицата под LED се наоѓаат копчињата МОТОР, АРР и РРР што се виртуелни еквиваленти на BTNM, BTNMz и BTNMu соодветно, а над копчето МОТОР е виртуелниот еквивалент на копчето BTNONOFF.

ПВИ „loop frequency“ (слика 2.19 (13)) пресметува колкупати кодот се извршува во една секунда и истото го прикажува до индикаторот на долниот дел на контролната табла.

На долниот дел на КТ се сместени контролери со кои операторот може да ги промени дифолт вредностите за програматорската матрица која се користи од страна на ПВИ `programator.vi`, промена на ФИ што е вредност за која се врши корекција на факторот на исполнетост при значително отстапување на тековната од потребната брзина на вртење и за вредноста на епсилон со која се дефинира дозволения интервал за брзината на вртење кои се користат од страна на ПВИ `driverM.vi`. Системот функционира задоволително со дифолт вредностите, па промените во овој дел не се неопходни. Во долниот десен агол на КТ се индикаторите за тековната вредност на факторот на исполнетост, како и индикаторот за грешка.

Во долниот лев агол на КТ се наоѓа копчето за исклучување на програмата, СТОП, со чие притискање се излегува од главниот круг и контролерот за серискиот порт на кој е приклучено Arduino-то, што мора да биде точно зададен за програмата успешно да стартува.

Во случај да настане грешка при комуникација со хардверот или се притисне копчето СТОП, програмата ќе излезе од главниот круг, ќе ги исклучи LED, сирената и моторот како што е прикажано на (слика 2.19 (17)) – доколку се работи за регуларно исклучување без грешка, па ја затвора комуникацијата со Arduino-то преку ПВИ означен со (18) на слика 2.19.

Првичнава верзија на програмата беше тестирана и не се јавија нелогичности при нејзиното извршување, па со ова може да се каже дека софтверското решение е успешно изведено.

Бојан Петрески