

СОФТУЕРНА РАЗРАБОТКА НА СИСТЕМА ЗА МИКРОВЪЛНОВО ИЗМЕРВАНЕ НА СКОРОСТТА НА ДВИЖЕЩИ СЕ ОБЕКТИ

Христо Килифарев¹, Борислав Недев¹

¹ Технически университет - Габрово

SOFTWARE DEVELOPMENT OF A SYSTEM FOR MICROWAVE SPEED MEASUREMENT OF MOVING OBJECTS

Hristo Kilifarev¹, Borislav Nedev¹

¹ Technical University of Gabrovo

Abstract

This paper presents the development of software for a microcontroller from the Arduino platform, controlling the processes in a system for measuring the speed of moving objects using a microwave radar (switch) RCWL-0516. In the scheme of the latter, modifications have been made, through which it becomes possible to derive a digital signal corresponding to the doubled Doppler frequency, which depends on the speed of the moving object in the range of the sensor. The main goals of the software are to automate the processes of measuring and analysing the pulses received by the modified microwave sensor, data processing, visualization and user interface implementation. General algorithms of operation are presented, a suitable Arduino controller is selected according to the defined functionalities and requirements of the measurement system. Results of tests conducted to prove of its operability and functionality of the system are presented.

Keywords: development of software; Arduino; measuring the speed; microwave radar (switch) RCWL-0516.

ВЪВЕДЕНИЕ

Често измерването на скорост на подвижни обекти с микровълнов радарен сензор се свежда до определяне на параметъра – доплерова честота. Тя се получава, когато излъчените от сензорния модул микровълни се отразят от движещ се обект в работния му обхват. Приближаването към или отдалечаването от сензора, съответно прибавя или изважда стойност към отразената вълна, пропорционална на скоростта на обекта. Тази стойност е известна като доплерова отместване и зависи също от излъчваната честота от сензора.

Модулът микровълнов радар (ключ) RCWL-0516 [1] фабрично не е пригоден за измерване на скорост на движещи се обекти, а само да регистрира присъствие и да отработи висок изходен потенциал (3,3V). Използва се обикновено за управление на LED осветителни тела. В структурата си той съдържа интегрална схема RCWL-9196,

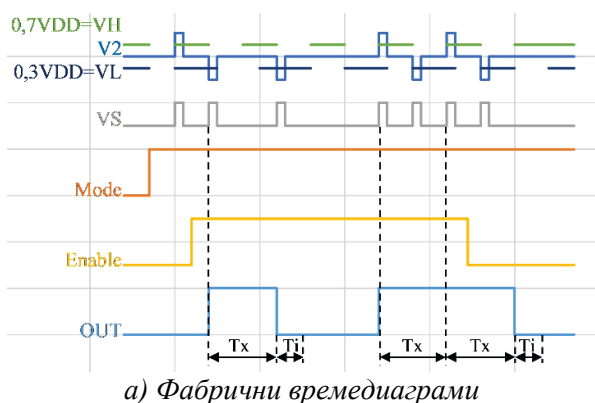
чиято цел е да генерира изходен импулсен сигнал в съответствие с входен аналогов сигнал с доплерова честота.

За да е възможно сензорният модул RCWL-0516 да се използва за измерване на скорост се извършва модификация в схемата му по такъв начин, че на изхода му се извежда импулсен сигнал, който съответства на удвоената доплерова честота.

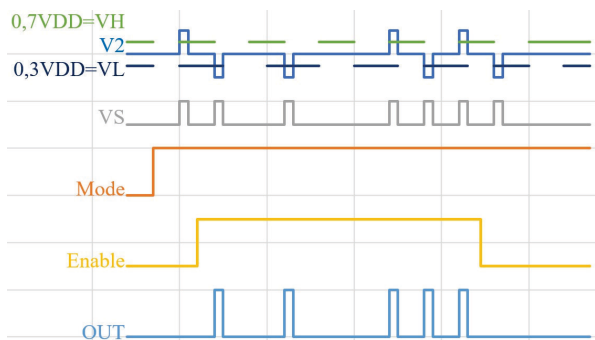
След обработка и анализ на изходните импулси от сензорния модул се извежда информация за измерената скорост по подходящ начин на дисплей.

За стартиране на измерване на скоростта от системата се използва бутон, като измерванията се извършват циклично и се визуализират, докато той е натиснат. Заложените функции в системата могат да се реализират с управление, базирано на подходящ Arduino контролер.

В настоящия доклад се представя разработването на програмно осигуряване на



а) Фабрични времедиаграми



б) След модификация в схемата

Фиг. 2. Времедиаграми за работата на РСМН 0106

Изработените импулси от схемата на извод 2 (OUT) ще бъдат с 2 пъти по-висока честота от доплеровата, поради факта, че „отрицателните“ импулси на V2 (фиг. 2б) влизат в сигнала VS [3].

Компараторите CP1 и CP2 (фиг. 1) сравняват нискочестотния сигнал от операционен усилвател OP2, свързан като активен лентов филтър в радарния модул RCWL – 0516. Той получава сигнал от OP1, свързан също като активен лентов филтър. Двата филтъра имат еднакви честоти на срязване като честотната им лента е между 0,5 Hz и 16 Hz.

Чрез замяна на кондензатори и резистори в схемата на OP1 и OP2, може да се разшири честотния обхват и усилването на тези филтри, което ще позволи да се измерва по-висока доплерова честота (съответно по-голяма скорост), а също и да се приемат сигнали с по-малко ниво, което ще увеличи работния обхват на сензора, т.е. ще се измерват по-малки или по-отдалечени подвижни цели.

Ако от (1) се изведе, че радиалната скорост на движещия се обект е (3):

$$U = f_D \cdot \lambda / 2, \quad (3)$$

то при отчитане на измерваната честота f_M (4):

$$f_M = 1 / T_M = 2 \cdot f_D \quad (4)$$

получаваме, че скоростта може да се изрази като (5):

$$U = f_M \cdot \lambda / 4, \quad (5)$$

което отнесено към измервания период T_M води до (6):

$$U = \lambda / (4 \cdot T_M). \quad (6)$$

Отчитайки (2), получаваме, че скоростта на обекта може да се изчисли чрез (7):

$$U = \lambda / (4 \cdot T_M) = 0,023585 / T_M \quad (7)$$

За увеличаване на точността на измерваната скорост се осредняват няколко поредни периода в изходния сигнал. При осредняване на четири поредни периода, това се изразява с (8):

$$T_m = (T_{M1} + T_{M2} + T_{M3} + T_{M4}) / 4. \quad (8)$$

Преобразуването на изчислената скорост от формула (7) от мерни единици [m/s] в [km/h] става чрез формула (9):

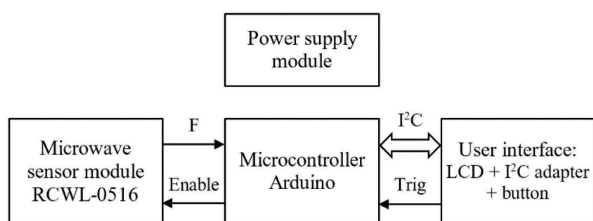
$$U_{kmh} = U \cdot 3600 / 1000. \quad (9)$$

Б. Функционална блокова схема и избор на компоненти за реализация

На фиг. 3 е представена функционалната блокова схема на системата за измерване на скорост, базирана на модифицирания микровълнов радарен сензорен модул RCWL-0516.

За прототипната реализация на системата като управляващ модул е избрано да се използва контролер Arduino UNO, който има достатъчно хардуерни ресурси за конкретното приложение и много допълнителни ресурси за бъдещо разширяване на функционалността на системата [4].

Друга важна причина за избора на този контролер е наличието на вграден стабилизатор на напрежение в платката му, който дава възможност за външно захранване с напрежения от 7 до 15 V DC. Избран е



Фиг. 3. Функционална блокова схема на системата за измерване на скорост

адаптер за мрежово напрежение 220V AC/12V DC за захранване на Arduino UNO контролера. Чрез изводите му Vcc и GND се захранват другите компоненти в системата със стабилизирано напрежение 5V DC. По този начин се реализира захранващият модул във функционалната блокова схема.

За работата на системата е предвидено да се използва потребителски интерфейс, реализиран чрез стандартен 16x2 LCD модул (с 2 реда и 16 символа на ред), свързан към Arduino UNO чрез с I²C интерфейсен адаптер [5] и един бутон. На дисплея ще се извежда актуална информация за текущата измерена скорост на движение на обект в работния обхват на сензора, докато бутонът е натиснат, а когато не е натиснат – ще се визуализира последната измерена стойност на скоростта.

На фиг. 4 е представен проект за разполагане на извежданите данни на дисплея, а значенията на променливите са:

- ssss = "Curr" – при визуализиране на текущи измервания;
- ssss = "Last" – при визуализиране на последното измерване;
- XXX.X – измерената скорост в [m/s];
- YYY.Y – измерената скорост в [km/h].

В. Алгоритъм на работа и програмна реализация

На фиг. 5 е представена блокова схема на главния алгоритъм на работа на Arduino контролера, в който началото съответства на момента на стартиране на програмното осигуряване след рестарт или включване на захранването. Еднократно се извършва инициализиране на всички използвани изводи

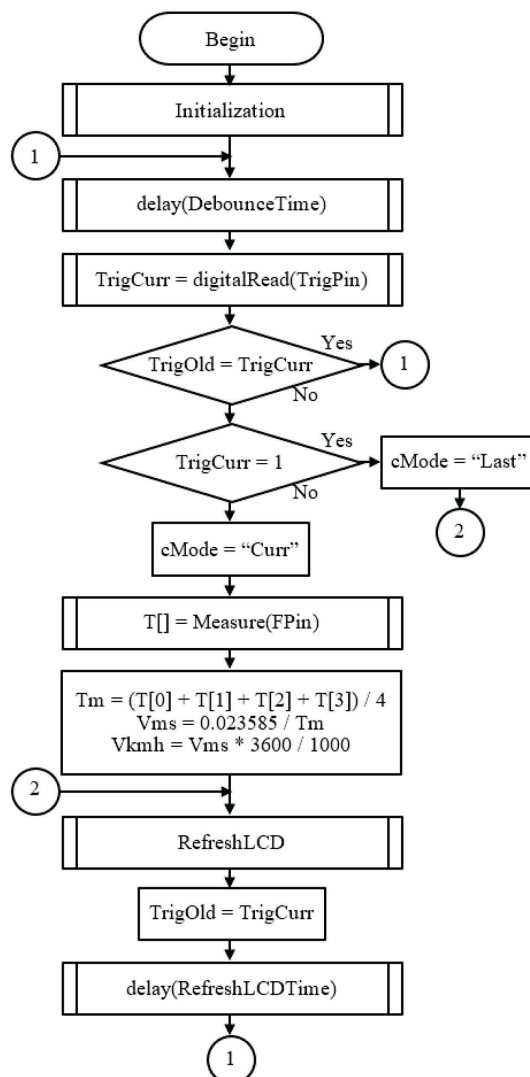
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Row 0	S	p	e	e	d			m	/	s	:	X	X	X	.	X	
Row 1	s	s	s	s	s			k	m	/	h	:	Y	Y	Y	.	Y

Фиг. 4. Проект за разполагане на извежданите данни на дисплея (LCD 16x2)

на контролера, комуникацията по сериен интерфейс за диагностични цели и LCD модула, към който се извежда начален екран с ssss = "Last" и скорост 0. Активира се също изтеглящ резистор към Vcc за извода TrigPin, към който е свързан бутоната.

Следващата част от главния алгоритъм се изпълнява циклично до изключване на захранването или рестартиране.

С функцията delay(DebounceTime) се изчаква зададено време (около 20 ms) за филтриране трептенето на контактите на бутоната, след което с извикването на функция TrigCurr = digitalRead(TrigPin) се прочита текущото състояние на извода TrigPin, към който е свързан бутоната и се съхранява в TrigCurr. Проверява се дали има промяна в състоянието на бутоната чрез сравняване с неговата стара стойност TrigOld. Ако няма промяна, алгоритъмът продължава с ново изчакване, а ако има такава се проверява,



Фиг. 5. Блокова схема на главния алгоритъм на работа

каква е новата му стойност. Ако $TrigCurr = 1$, тогава $cMode = "Last"$ и действието на алгоритъма продължава към преход 2, а в противен случай бутонът е натиснат и се преминава в режим на измерване – $cMode = "Curr"$.

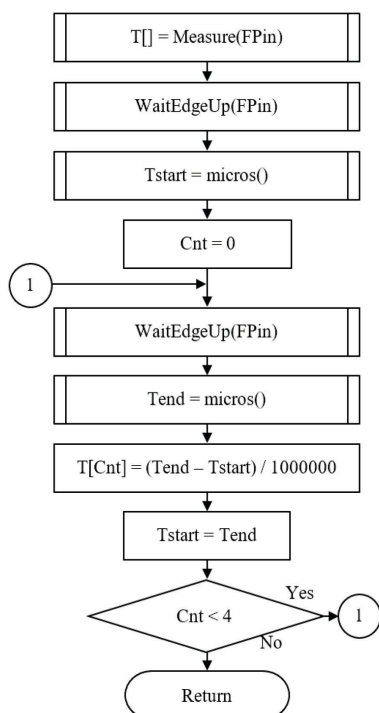
Следва извикване на функцията $Measure(FPin)$, извършваща 4 поредни измервания на периода на сигнала от сензорния модул, постъпващ на извод $FPin$, като записва стойностите в масив T (фиг. 6). След приключване на измерванията се извършва осредняване на четирите измерени времеви интервали по формула (8) и се записват в променливата Tm . Пресмята се скоростта на обекта Vms в [m/s] по формула (7) и $Vkmh$ в [km/h] по формула (9).

В края на главния алгоритъм се извършва опресняване на извежданата информация към LCD за променливите $cMode$, Vms и $Vkmh$, които съответстват на $ssss$, $XXX.X$ и $YYY.Y$ според проектирания шаблон за разполагане (фиг. 4). Преди извеждането на стойностите за скорост е нужно да се направи подходящо преобразуване на дробните числа във формат с един разряд след десетичната точка и във вид на низ от символи (фиг. 7). След това се съхранява текущото състояние на бутона като негово старо ($TrigOld = TrigCurr$), за да може да се засече промяна в състоянието му. Изчаква се зада-

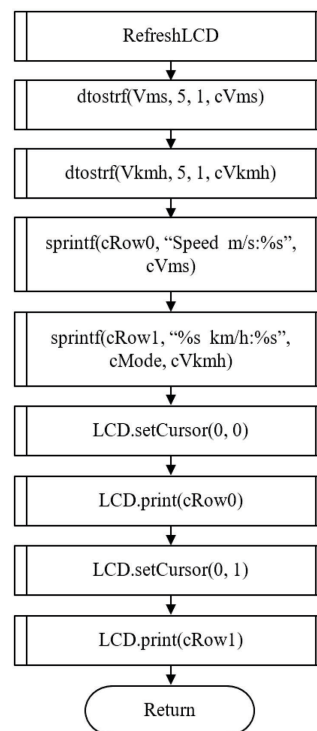
дено време за опресняване на LCD (около 500 ms), след което действието продължава с повторението на цикличната част – преход 1 (фиг. 5).

Важно да се отбележи, че тялото на алгоритъма на функцията за измерване на входния сигнал от сензорния модул (фиг. 6) се изпълнява 4 пъти чрез броячът Cnt , който се инкрементира с единица при всяко повторение на цикъла. По този начин Cnt се използва като индекс в масива $T[Cnt]$, в който при всяка итерация се записва текущата измерена стойност на периода на сигнала. Това става като се използва функцията $WaitEdgeUp(FPin)$, която изчаква постъпването на нарастващ фронт, а когато това се случи, чрез функцията $micros()$ се отчита текущата стойност на независим външно синхронизиран с тактовата честота брояч в микросекунди [6]. Разликата между две поредни засичания на нарастващия фронт на сигнала ($Tend - Tstart$) представляват периода му. Тя се преобразува в секунди, чрез деление на 1000000, след което се записва в текущата клетка от масива $T[Cnt]$. В края на цикъла $Tstart = Tend$, за да се изчисли следващия период по същия начин.

На фиг. 7 е представен алгоритъмът на работа на функцията $RefreshLCD$. Тя преобразува измерената скорост и режима на



Фиг. 6. Блокова схема на алгоритъма за измерване на периоди в сигнал F



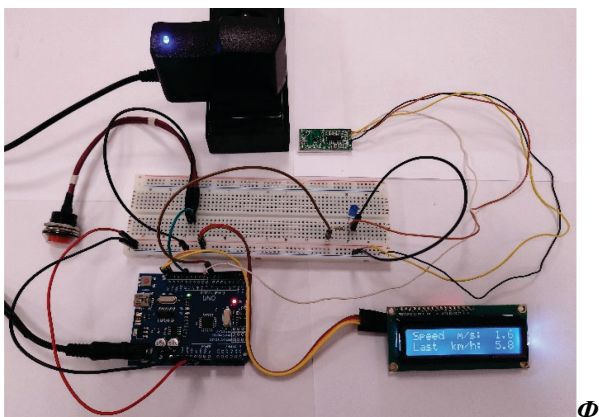
Фиг. 7. Блокова схема на алгоритъма за извеждане на актуални стойности към LCD

работа във форматирани символни низове (фиг. 4) и ги извежда на LCD.

За реализацията на бързо прочитане на входовете на Arduino контролера се използва външната функция `digitalReadFast(FPin)` от библиотеката `digitalWriteFast.h` [7], която работи чрез директно обръщение към портовете, постигайки голямо бързодействие.

Г. Реализация и резултати

Фигура 8 е снимка на прототипната реализация на системата. LCD визуализира последната измерена стойност на скоростта при движение на обект в работния обхват на микровълновия сензор.



иг. 8. Прототипна реализация на системата за измерване на скорост на обекти

ЗАКЛЮЧЕНИЕ

Направените експерименти и постигнати резултати показват, че системата е работоспособна.

Точността на извежданите стойности на измерена скорост са непотвърдени експериментално. Те са теоретично изчислени по представените в доклада формули. За да се използва разработената система като измервателен инструмент е нужно тя да се тества допълнително в лабораторни условия със специализирана постановка, в която движещият се обект се премества с известна постоянна скорост, спрямо сензорния модул. След провеждане на достатъчно изследва-

ния може да се докаже достоверност на измерените скорости с определена точност.

Заложените функционални възможности в системата са реализирани със сравнително евтини компоненти, даващи възможности за бъдещо разширяване. По този начин разработката може да намери приложение в автоматика, роботика, транспорт, системи за сигурност и др.

БЛАГОДАРНОСТИ

Този доклад е финансиран по проект № 2304Е „Повишаване на ефективността на обучението чрез използване на информационни и комуникационни технологии“ за научни изследвания на Технически университет – Габрово.

REFERENCE

- [1] SUNLEPHANT, Microwave Radar Proximity Sensor RCWL-0516, Datasheet, <https://www.mantech.co.za/datasheets/products/RCWL-0516.pdf>, 2023.
- [2] Wolff, Ch., Radar Basics, Continuous Wave Radar, <https://www.radartutorial.eu/02.basics/Continuous%20Wave%20Radar.en.html>, 2023.
- [3] Desbonnet, Joe, RCWL-0516 information, 27 Mar 2023, <https://github.com/jdesbonnet/RCWL-0516>, 2023.
- [4] Arduino, Boards, <https://store.arduino.cc/collections/boards>, (2023).
- [5] DFRobot, I2C 16x2(1602) LCD Display Module for Arduino, <https://www.dfrobot.com/product-135.html>, (2023).
- [6] Arduino, Reference > Language > Functions > Time > Micros(), <https://www.arduino.cc/reference/en/language/functions/time/micros/> (2023).
- [7] github.com, Watterott, Armin Joachimsmeier, digitalWriteFast, <https://github.com/ArminJo/digitalWriteFast>, (2023).