

e-ISSN: 2582-5208

## International Research Journal of Modernization in Engineering Technology and Science

( Peer-Reviewed, Open Access, Fully Refereed International Journal ) Volume:05/Issue:10/October-2023 Impact Factor- 7.868 w

www.irjmets.com

# APPLICATION OF ARDUINO UNO AS AN AUTOMATIC CURTAIN OPENING AND CLOSING SYSTEM CONTROLLER

### Yessy Dea Sylvia Saragih<sup>\*1</sup>, Sarah Karima Indah<sup>\*2</sup>, Susaidin<sup>\*3</sup>

<sup>\*1,2</sup>Politeknik Negeri Medan, Teknik Elektro, Teknik Listrik, Medan, Indonesia.

<sup>\*3</sup>Politeknik Negeri Medan, Teknik Elektro, Teknik Elektronika, Medan, Indonesia.

DOI: https://www.doi.org/10.56726/IRJMETS45653

### ABSTRACT

This research was carried out with the aim of simplifying the programming design of making an automatic curtain system based on Arduino Uno by creating a programming language that will be used. From this research, an Arduino Uno programming language was obtained to control the automatic curtain opening and closing operating system. Software design stages in the Arduino Uno programming language using C language and registers from Arduino. As the main program in this system, the C language is used to design functions on Arduino which play a major role in managing the functions of the system.

Keywords: Arduino Uno, Programming Language, C Language.

#### I. INTRODUCTION

This research uses an RTC module which functions as a time and date reminder system which uses a battery energy source to keep the system running. Apart from that, this module can also update the date and time periodically. This design uses an adapter with a voltage of 12 volts to input the Arduino and also as a source for the motor. The adapter used has a principle like a full wave rectifier with a bridge system. This system is designed in two conditions, namely manual (Remote) and automatic (RTC). Arduino is not just a development tool, but a combination of hardware between a programming language and a developing Integrated Development Environment (IDE). IDE is a software that has a big role in writing programs, combining them regularly into a binary code language and uploading them into a microcontroller memory. (Fezari & Al Dahoud, 2018). In designing the tool, there is hardware and software that will be used, namely:

1. Hardware > in/out board (Input(I)/Output(O))

2. Software > Arduino software is an IDE that functions to write programs and to supply connections to computers. For example, it can be seen from programs and storage for program development.

Arduino Uno is a microcontroller board based on the ATmega328 datasheet (Fathulrohman & Saepulloh, 2019). In this project, Arduino Uno is used as input in the form of a remote control and output in the form of a DC motor, where if the Remote Control interface gives a command to the Arduino then the Arduino executes the command. into the output in the form of a voltage, which voltage is used to activate an electronic relay that is connected to control the left or right rotation of the load (motor).

#### II. METHODOLOGY

The block diagram of the working method of this program is as follows:



Figure 1: Block diagram of curtain control design with remote and RTC

The design of opening and closing curtains automatically and manually is deliberately designed to facilitate activities related to percentages with two setting positions, namely automatic and manual. The automatic driver is Arduino, RTC and electronic relay, while the manual is Remote Control.

The main parts of the system consist of:

1. Arduino Uno ATMega 328

www.irjmets.com



e-ISSN: 2582-5208

#### International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:05/Issue:10/October-2023 Impact Factor- 7.868

www.irjmets.com

- 2. Electronic Relay
- 3. Remote and Receiver
- 4. RTC (Real Time Clock)
- 5. DC motors

6. MCB

The software design stages on Arduino use C language and registers from Arduino. As the main program in this system, C language is used to design functions on Arduino which play a big role in regulating the function of the system. In determining the safety rating required in the final project. Make coding on the Arduino Uno

- 1. Open the Arduino software (application), until it appears like the picture.
- 2. Then create a program that we will use in the Arduino Application.

<pre>k Eff Stach Toek Hep T. Qostin, funnais T. Dostin, funnais T. Dos</pre>	X	D
TA_Gooder_connais include GUILA by include GUI		
TA_CodeW_UTINE TainLike GTIILS tainLike GTIILS tainLike GTIILS SESINF FIT: SESINF FIT: SESINF FIT: SESINF FIT: SESINF FIT: SESINF TO SESINF TO SE	D	
TA_Code_connects Final define (defined as pin #7// define GM # 7//KH is defined as pin #7//KH is defined is pin #7//KH is defined is pin #7//KH is defined GM # 7//KH is defined GM # 7/		
<pre>include GCTLin b. GT_DSIGNT FIC; Meffine GM 7 //CM is defined as pin #7// define CM 8 //CM is defined as pin #4// int set_smallhup=10; int set_smallhup=10; int set_smallhup=20; def setup 01 //Setup runs once// setup setup.020; def setup 01 //Setup runs once// setup setup.020; def setup 01 //Setup runs once// setup setup.020; def setup 01 //Set CM as an output// NERN Rep101; SETU Rep101; setup 01 //Loop runs forever// Define non-REC.cov(): setup 01 //Loop runs forever// Define non-REC.cov(): setup 01 //Loop runs forever// Define non-REC.cov(): setup 0. setup 0</pre>		
<pre>include GDE.m.bo MC_SIGNO FIT: SetLing CDT 7 //GY is defined as pin #7// Hertime COT # //CX is defined as pin #7// is set_publication = 0; is set_publication = 0; indebe COT, COTTON : //Set COT as an output// indebe COT as forever// indebe COT, COTTON : //Set COT as an output// indebe COT, COTTON : //Set COT as an output// indebe COT as forever// indebe COT</pre>		
<pre>Mt_DSIANP ETC; MtfLex CVT //CVI is defined as pin #7// Mtefine CVT # //CVI is defined as pin #7// Mtefine CVT # //CVI is defined as pin #7// mte det_landbase; mte det_landbase; mte det_landbase; mtef_landbase;</pre>		
<pre>Methics CM 7 //CM is defined as pin #7// Methics CDM 8 //CDM is defined as pin #7// miss set_pathore_=DS; miss set_pathore_=DS;</pre>		
<pre>Mediae COM # //COM is defined as pin 40// iii set_painFutup &lt;10; iii set_painFutp &lt;10; iii set_painFutp &lt;10; iii set_painFutp &lt;10; i</pre>		
<pre>st: set_jumDitup -10; if set_jumDita+0; if set_jumDita+0; if set_pumDita+0; if set_pumDita+0; if</pre>		
<pre>sis set_setUseUse_sis is set_setUseUse_sis is set_setUseUse_sis is set_setUseUse_sis is set_setUseUseUseUseUseUseUseUseUseUseUseUseUseU</pre>		
<pre>mix set_pusition-5; ist set_pusitive-5; ist set_pusitive-02; ist se</pre>		
<pre>sis set_meinTurneyOC; end setup() [ //Setup runs once// setal.setup() [ //Setup runs once// setal.setup() [ //Set OH as an output// indede(OK, OTTUT) //Set OH as an output// Kets.setup() [ ETC.setup(); ETC.setup(); setal.print(own)set(), EC); setal.print(own)set(), EC); Setal.print(own)set(), EC); Setal.print(own)set(), EC); Setal.print(own)set(), EC); Setal.print(own)set(), EC); Setal.print(own)set(), EC);</pre>		
<pre>sis set_mellinesped2; weid setup() { //Setup rums conce// setal.begin(960); Minch egg(); Minch egg(); Minch egg(); setal.print(); setal.print(); setal.print(); setal.print(); Setal.</pre>		
<pre>wid setup() [ //Setup runs onco// errail.setup(setup); //Set CW as an output// indebs(CW, OUTCUT); //Set</pre>		
<pre>ieral.begin(exec_set)/ setal.print(v); Se</pre>		
<pre>index(0, 0TET); //Sec COM as an output// index(COM, 0TET); //Sec COM as an output// Mint hepin(); ind loop() (//Loop runs forewar// BateFine now-RCL.cow(); secal.print(now.year(), ECD); Serail.print(now.year(), ECD); Serail.print(n'); Serai</pre>		
<pre>indeds(0x), 0TFEUD;://Set COE as an output// Kits.hegin();  PEC.hegin();  exist long() ( //Loop runs forever// BateTime non-RET.cov():     son=RET.cov():     son=RET.cov():     son=RET.cov();     Serial.print(r/r):     Seria</pre>		
<pre>Kirs.Acgin(); BDC.hegin(); MadeTime now=RUT.cow(); Serial.print(nowsymar(), HED; Serial.print(nowsymar(), HED; Serial.pri</pre>		
<pre>RTC.hegis(); end loop() [ //Loop runs forewar// DateHim som=RTC.now(); som=RTC.now(); Serial.print(over,mail(), HEC); Serial.print(over,mail(), HEC); Serial.print(over,mail(), HEC); Serial.print(over,mail(), HEC); Serial.print(over,mail(), HEC); Serial.print(over,mail(), HEC); Serial.print(over,mail(), HEC);</pre>		
<pre>dellog() ( //Log runs forever// ButeTues now=RTC.now(); som=RTC.now(); som=RTC.now(); Serial.print(//'); Serial.print(now.not(), EC); Serial.print(now.not()</pre>		
<pre>wid loop() {//Loop runs forever// BateFine now=RC.now(): Serial.print(now.year(), EC): Serial.print(now.year(), EC): Serial.print(now.nown1(), EC): Serial.print('): Serial.print('): Serial.print('): Serial.print('): </pre>		
<pre>BetFime now-RUL.cov(): now-RUL.cov(): Serial.print(r/): Serial.print(r/): Serial.print(r/): Serial.print(r/): Serial.print(r/): Serial.print(r/): Footst note foot durft INT()</pre>		
<pre>som=ETC.how(); Serial.print(pow.par(), EC); Serial.print(print()'); Serial.print(print(print(), EC); Serial.print(print(), EC); Serial.print(print(), EC);</pre>		
Serial.print(norsystef), ECD; Serial.print(norsystef), ECD; Serial.print(norsystef), ECD; Serial.print(''); forsit series form form for the series of the se		
Serial print(//): Serial print(wosth(), EC); Serial print(//): Fordat works from doubt, TEC:		
Serial prior (non-month), EEC; Serial prior (non-month), EEC; Endixt mades from durity INFA:		
Serial protection from durity . They		

#### Picture 1. Listing Program on Arduino

3. Compile coding on the Arduino Uno, the aim is to check whether the program that has been filled in is correct or not. And then click the check mark/verify in the Arduino Uno software.

4. After clicking the button, wait a few moments until the checking process is complete, then Done complling will appear which indicates the checking process has been completed, if the program is correct it will appear as shown in the following image:

<pre>publication [W, UUEU]; //Set CW as an output// publication [V]; [/Set CW as an output// Wire.begin[); RTC.begin[); } wid loop() { //Loop runs forever// Datefine now=RC.now(); now=RC.now(); serial.print(now.year(), TEC); serial.print(now.year(), TEC); serial.print(now.noth(), DEC); serial.print(now.noth(), DEC); Serial.print(now.noth(), DEC); Serial.print('/'); Exercl uses field bytes (138) of program storage space. Maximum is \$2256 bytes.</pre>	
<pre>publick [CDK, OUTENT); //Set CDW as an output// Wire.legin(); Wire.legin(); Widdog() { //Loop runs forever// DateFine now=RC.now(); now=RC.now(); Serial.print(now.year(), IEC); Serial.print(now.year(), IEC); Serial.print(now.nonth(), IEC); Serial.print(/'); Exerci under form dow(), IEC); Serial.print(), IEC); Serial.print(),</pre>	
<pre>Wink.begin(); Wink.begin(); Wink.begin(); Wink.begin(); Wink.begin(); DateFine non-HUL.now(); DateFine non-HUL.now(); Serial.print(Now.year(), IEC); Serial.print(Y'); Serial.print(Y'); Exact</pre>	
<pre>ETC.hegin(); } void loop() { //Loop runs forever// DeteTime now=ETC.now(); serial.print(now.year(), DEC); Serial.print(now.year(), DEC); Serial.print(now.nonth(), DEC); Serial.print('/'); Exercit nowenthate doubt north. Dume compliag. Second pages 4446 bytes [138] of program storage space. Maximum is \$2256 bytes.</pre>	
] wid loo() { //Loop rms forever// DateTime nov=RC.now(); serial.print(low,yser(), DEC); Serial.print(low.nonth(), DEC); Serial.print('/'); serial.print('/'); terisl.print('/'); terisl.print('/'); Exercisl.print	
<pre>vvid loop() { //Loop rmms forever// DateFime now=RCL.now(); now=RCL.now(); serial.print(now.year(), REC); serial.print('/'); serial.print('/'); serial.print('/'); serial.print('/'); Exercit market here down() TET'. Dune compling. Sietch mass 4446 bytes [138] of program storage space. Maximum is \$2256 bytes.</pre>	
Datefine nov=RC(now(); nov=RC(now(); Serial.print(uow.year(), REC); Serial.print('/'); Serial.print('/'); Serial.print('/'); Control under former devint ("RE"). Dane compling.	
now=BTC.cow(); Serial.print(now.year(), DEC); Serial.print('/'); Serial.print('/'); Event() meter form dow() TEP): Dune compling. Sietch uses 4446 bytes (138) of program storage space. Maximum is \$2256 bytes.	
Serial.print(1/'); Serial.print(1/'); Serial.print(1/'); Serial.print(1/'); Earth and ther doubt TETL. Due compling.	
Serial.print(//); Serial.print(now.month(), DEC); Serial.print(//); Exactly under them doubt _ mPPL. Dune compling.	
Serial.print (now.month(), 100); Serial.print (//); Ential meter form down) - TOTA Dune compling. Sletch uses 4448 bytes (138) of program storage space. Maximum is 32256 bytes.	
Serial, print (//); Restil ander Acer deell TDPL. Dune compling. Sletch uses 4448 bytes (138) of program storage space. Maximum is 32256 bytes.	
Canada and Andrea Annalis 2001. 2001. Dune compiling Sletch uses 4448 bytes (138) of program storage space. Maximum is 32256 bytes.	
Dune compiling. Sletch uses 4448 bytes (138) of program storage space. Maximum is 32256 bytes.	۷
Sletch uses 4448 bytes (138) of program storage apace. Maximum is 32256 bytes.	
Sletch uses 4448 bytes (138) of program storage apace. Maximum is 32256 bytes.	
Sletch uses 4448 bytes (138) of program storage apace. Maximum is 32256 bytes.	^
Sketch uses 4440 bytes (13%) of program storage space. Maximum is 32256 bytes.	
Global variables use 398 bytes (198) of dynamic memory, leaving 1650 bytes for local variables. Maximum is 2040 bytes.	
	v
Attiving Genving Units on	омв

Picture 2. The display has finished compiling the program



e-ISSN: 2582-5208

# International Research Journal of Modernization in Engineering Technology and Science

( Peer-Reviewed, Open Access, Fully Refereed International Journal ) Volume:05/Issue:10/October-2023 Impact Factor- 7.868 ww

www.irjmets.com

5. The program is ready to be uploaded to Arduino Uno. Then upload the coding to Arduino with prepare the Arduino to be programmed; insert the Arduino USB port into the computer; Click the upload icon to upload the program to Arduino.

6. Arduino is ready to use.

# III. RESULTS AND DISCUSSION

From this research, test results were obtained in the form of data in the form of curtain travel time when testing closing and opening the curtains. The following are the measurement results of the data obtained:

MANUAL (REMOTE)							
DEMOTE	RELAY		CONDITION	RESPON			
REMUTE	1	2	CONDITION	REMOTE			
А	OFF	ON	CLOSED	DELAY 1,5 s			
В	ON	OFF	OPEN	DELAY 1,5 s			
С	-	OFF	CLOSED	DELAY 1,5 s			
D	OFF	-	OPENI	DELAY 1,5 s			

**Table 1.** Manual Testing of Opening and Closing Curtains

The table above is the test results from the manual remote control that we carried out. When button A is pressed, relay 1 will work and when button C is pressed it will work.

Table 2. Automatic Testing of Opening and Closing Curtains

OTOMATIS (RTC)				
SET WAKTU	KONDISI			
PUKUL 07.00 WIB	MEMBUKA			
PUKUL 17.00 WIB	MENUTUP			

In the table 2 are the results of the RTC testing that we carried out, we made a predetermined time setting, namely at 17.00 WIB the curtains will close automatically and at 07.00 WIB the curtains will open automatically.

#### **IV. CONCLUSION**

After completion of this research, the following conclusions can be drawn:

1. In a manual system, opening and closing the curtains always has a delay of around 1.5 seconds.

2. This curtain cover book can be applied in areas that are high and hard to reach, making things easier for people and making things more efficient and easier.

#### V. REFERENCES

- [1] Fezari, M., & Al Dahoud, A. (2018). Integrated development environment "IDE" for Arduino. WSN applications, 1-12.
- [2] Fathulrohman, Y. N. I., & Saepulloh, A. (2019). Temperature and humidity monitoring tool using Arduino Uno. Journal of Management and Information Engineering (JUMANTAKA), 2(1).
- [3] Haryanto, D., & Mulyani, S. (2022). Closing and Opening Curtains Automatically With Arduino Based Timing and Bluetooth Activation. JOURNAL OF INFORMATICS MANAGEMENT (JUMIKA), 9(1).
- [4] Nasir, J., & Difo, C. (2018). Application of Arduino Uno Based Color Blindness Test Tool. Symmetric: Journal of Mechanical Engineering, Electrical and Computer Science, 9(2), 925-934.