



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CAMPUS NATAL
CENTRO DE TECNOLOGIA
ENGENHARIA MECATRÔNICA
SISTEMAS DIGITAIS**

**ALLYSSON DE ANDRADE SILVA
ANNY BEATRIZ PINHEIRO FERNANDES
ISAAC DE LYRA JUNIOR
MARCELO FERREIRA MOTA JÚNIOR
STHEFANIA FERNANDES SILVA**

**COFRE DIGITAL BASEADO NO MICROCONTROLADOR
ATMEGA328P**

NATAL-RN

2021

Resumo

O ATMEGA328P é um microcontrolador amplamente utilizado em sistemas autônomos, principalmente, por ser barato e funcional. Diante disso, o seguinte relatório tem como objetivo desenvolver o projeto da abertura de um cofre digital usando o ATMEGA328P. Para isso, foi preciso entender o *datasheet* do microcontrolador, para assim definir os periféricos necessários no projeto e o *clock*. Além disso, foi preciso desenvolver uma lógica para o *display* e para o temporizador de meio segundo.

Palavras-chaves: ATMEGA328P; Microcontrolador; Arduino.

Lista de Figuras

1	Interface homem-maquina do sistema.	6
2	Elementos da interface homem-máquina.	7
3	Interface do cofre aberto.	7
4	Pinagem ATMEGA328P.	8
5	Diagrama de blocos ATMEGA328P.	9
6	Diagrama de Blocos - MDE	10
7	Legenda - Diagrama de Blocos - MDE	11
8	Estado Compara Senha	13
9	Exemplo de Comparação	14
10	PORTB - Registro de dados	15
11	Funções alternativas referente aos pinos do PORTB	16
12	Funções alternativas referente aos pinos do PORTC	16
13	Funções alternativas referente aos pinos do PORTD	17
14	Mapa de pinos do ATMEGA328P	17
15	Entradas PB6 e PB7	19
16	Registrador ADMUX	19
17	Bits seletores de referência	20
18	Registrador ADCSRA	21
19	Registrador ADCSRB	21
20	Registradores ADCH/ADCL	22
21	Diagrama de blocos da conversão para os displays	23
22	Tabela de divisões do <i>clock</i> interno	25
23	Horários das reuniões	30

Lista de Tabelas

1	Chave seletora e saída dos multiplexadores	24
2	<i>Duty cycles</i> para cada pino dos terminais do LED RGB	24
3	Identificação da equipe	28

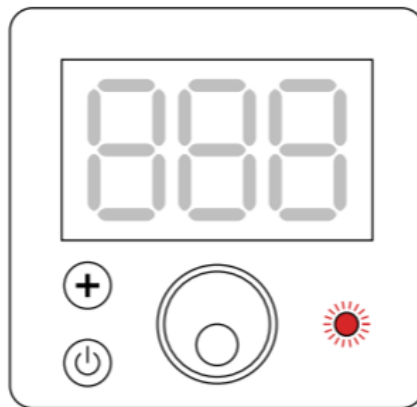
Sumário

1	INTRODUÇÃO	6
2	DESENVOLVIMENTO	8
3	PROJETO	10
3.1	DIAGRAMA DE BLOCOS - MDE	10
3.2	DETERMINAÇÃO DOS PINOS	14
3.3	CONVERSÃO A/D	19
3.4	Regulagem dos valores a partir do potencímetro	22
3.5	<i>DISPLAY</i> E <i>LEDS</i>	23
3.6	TEMPORIZADOR DE 0,5 SEGUNDOS	24
	REFERÊNCIAS	26
	ANEXO A - Relato semanal	28
A.1	Equipe	28
A.2	Defina o problema	28
A.3	Registro de <i>brainstorming</i>	28
A.4	Pontos-chave	29
A.5	Questões de pesquisa	29
A.6	Planejamento da pesquisa	30

1 INTRODUÇÃO

O microcontrolador ATMEGA328P é amplamente utilizado em projetos e sistemas autônomos de baixo custo. Diante disso, o seguinte trabalho tem como objetivo o desenvolvimento do projeto de um circuito que permita o controle da abertura da porta de um cofre, utilizando o ATMEGA328P. Tal projeto precisa ter embasamento suficiente para um código fonte em *Assembly* possa ser construído, além de necessitar de uma interface tal qual a mostrada na Figura 1.






Figura 1: Interface homem-maquina do sistema.



Fonte: Dados do problema.

O sistema possui entradas digitais e analógicas, onde os botões de iniciar/cancelar e de adicionar são digitais e o potenciômetro é uma entrada analógica voltada para definir o valor da senha inserida no cofre. Nesse sentido, a interface utiliza dos *displays* para indicar que o sistema está desligado, para exibir a senha inserida e para indicar que o cofre está aberto. Para o primeiro caso, todos os *leds* dos três *displays* ficam apagados, para o segundo caso será exibido o valor definido pelo potenciômetro de entrada e para o terceiro caso uma configuração específica é mostrada nos *displays*. Além disso, cada processo possui o acendimento de um *led* em particular. A Figura 2 resume o que foi dito.

Figura 2: Elementos da interface homem-máquina.

Elemento	Descrição
	Display para exibição de valor entre 0 e 999
	Potenciômetro para ajuste de valor (volta completa)
	Botão de inicializar/cancelar o processo (tipo <i>pushbutton</i>)
	Botão de adicionar valor (tipo <i>pushbutton</i>)
	Led RGB (Fechado; Senha; Processando; Aberto)

Fonte: Dados do problema.

Com respeito aos processos do sistema, quando o cofre está inativo o *led* aceso é o vermelho, para ser possível inserir as senhas o botão de iniciar deve ser pressionado. Na interface de inserção de senha o *led* aceso é o azul, na inserção de qualquer uma das três senhas pode-se cancelar o procedimento ao pressionar o botão iniciar/cancelar. Quando a senha inserida está sendo processada o *led* aceso é o laranja, esse processamento tem a duração de meio segundo e durante ele não pode ser feito o cancelamento da abertura do cofre. Por último, quando o cofre está aberto o *led* aceso é o verde e o *display* se comporta conforme mostrado na Figura 3.

Figura 3: Interface do cofre aberto.



Fonte: Dados do problema.

É válido salientar que o cofre somente será aberto se todas as 3 senhas estiverem certas e na ordem correta. Caso isso não seja atendido, o cofre volta ter o *led* vermelho aceso e os *displays* são apagados. Diante disso, o presente relatório tem como objetivo desenvolver o projeto do sistema explicado usando como base o ATMEGA328P.

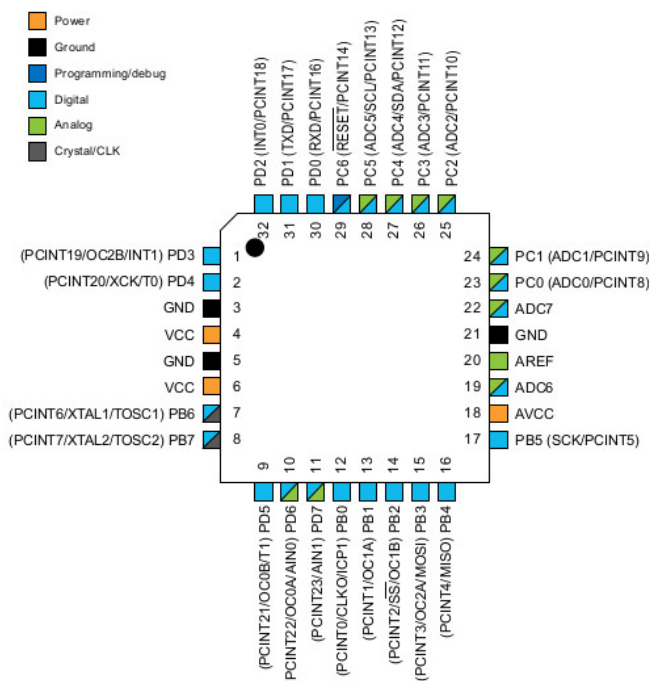
2 DESENVOLVIMENTO

A busca por um dispositivo que pudesse ser barato, funcional e facilmente programável motivou a criação do Arduino em 2005. A partir disso, também surgiu o conceito *hardware* livre, o qual é definido como um *hardware* básico que pode ser modificado, personalizado ou montado por qualquer um, até mesmo por estudantes sem muita experiência (THOMSEN, 2014).

Dessa forma, a placa do Arduino é formada por um microcontrolador Atmel e circuitos de entrada e saída, além de possibilitar uma conexão ao computador via USB (THOMSEN, 2014). Dentre as placas mais utilizadas há o Arduino Uno, que possui como núcleo o microcontrolador o ATMEGA328P (PÉREZ, 2018).

Dado o enfoque do relatório em projetar um circuito pautado no ATMEGA328P, uma ênfase maior será dada a esse microcontrolador. Conforme mostra a Figura 4, este possui 32 pinos onde 23 são entradas/saídas (digitais ou analógicas) programáveis, além de conter duas alimentações do sistema, uma do conversor analógico-digital (AD), três aterramentos (GND) e duas entradas a mais do conversor AD (além das entradas PC, há as entradas ADC6 e ADC7) (PÉREZ, 2018).

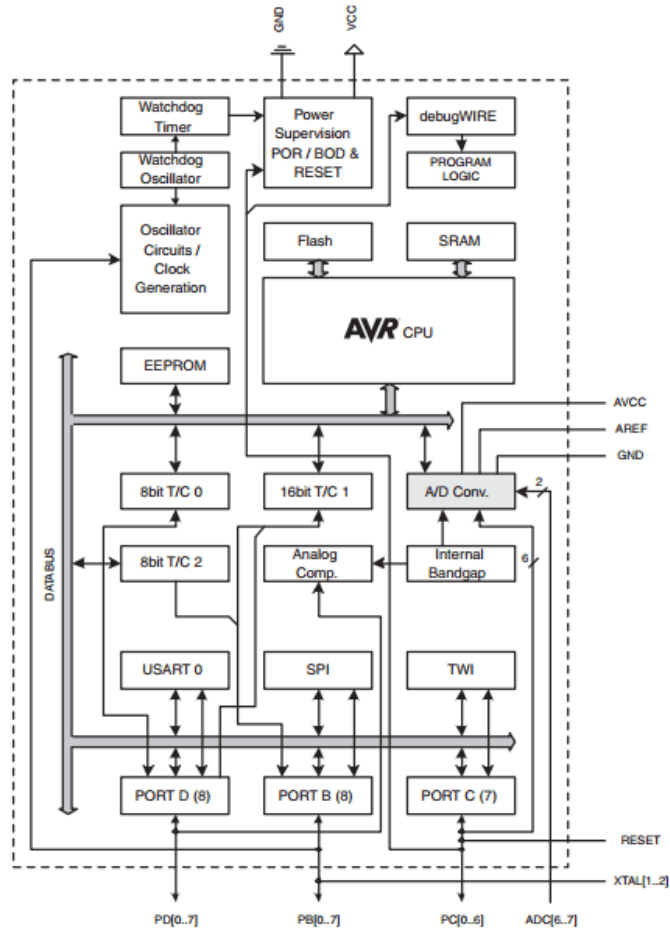
Figura 4: Pinagem ATMEGA328P.



Fonte: Amith Gopakumar.

O ATMEGA328P trata-se de um CMOS (*complementary metal-oxide-semiconductor*) de baixa potência com arquitetura RISC AVR avançada de 8 bits, o qual possui alto desempenho e baixo consumo. Na Figura 5 pode-se observar o diagrama de blocos desse componente.

Figura 5: Diagrama de blocos ATMEGA328P.



Fonte: Atmel (2016).

O componente possui uma memória SRAM de 2KB, memória flash de 32KB, memória EEPROM de 1KB e a frequência máxima de operação é de 20 MHz. Além de possuir 131 instruções, dois temporizadores/contadores de 8 bits e um de 16 bits, 6 canais PWM, interfaces seriais (SPI, USART e TWI), temporizador de vigilância, um comparador analógico e interrupções (PÉREZ, 2018).

A CPU conta com dois importantes componentes um banco de registradores 32x8 e uma ALU de 8 bits que opera com o conteúdo de dois registradores do banco de registros. Essas operações ocorrem um ciclo ou dois ciclos de *clock* (ATMEL).

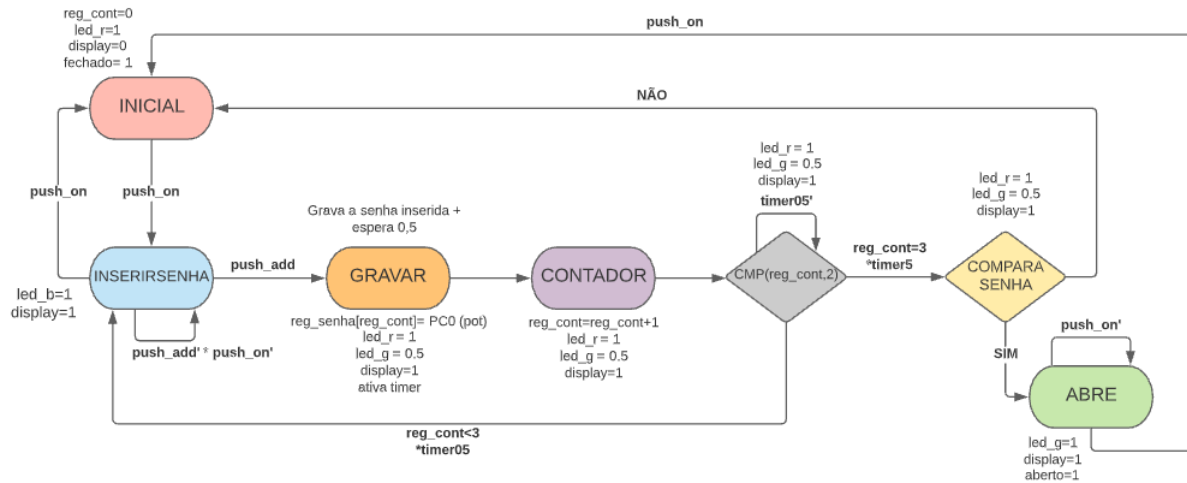
3 PROJETO

O projeto foi realizado definindo um fluxograma do procedimento que deve ser feito em *Assembly* para o cofre operar e definindo o uso dos periféricos do ATMGE328P. Os subtópicos a seguir irão explicar cada um desses procedimentos.

3.1 DIAGRAMA DE BLOCOS - MDE

Com intuito de retratar o funcionamento do cofre digital proposto e assim, projetar o algoritmo base para a posterior implementação em *Assembly* das funcionalidades requeridas, foi desenvolvido o diagrama de blocos da Figura 6.

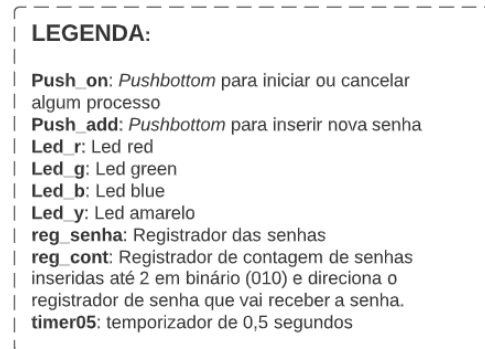
Figura 6: Diagrama de Blocos - MDE



Fonte: Elaboração dos Autores.

Este diagrama, legendado na Figura 7, possui sete estados, realiza uma sequência de *loops* para a inserção de senhas e ao fim verifica os valores inseridos para definir se realiza ou não a abertura do cofre.

Figura 7: Legenda - Diagrama de Blocos - MDE



Fonte: Elaboração dos Autores.

Este inicia seu funcionamento retratando o momento em que o cofre está trancado (fechado=1), com o display apagado (display=0) e com o LED aceso na cor vermelha (led_r=1). Este é o estado INICIAL e as variáveis iniciadas nele possuem o propósito de configurar o sistema para aquela funcionalidade. Com isso em vista, também é iniciada a variável reg_cont em 0 pois esta irá realizar a contagem de quantas senhas foram inseridas e também servirá como direcionamento para qual registrador de senha utilizar para armazenar a senha atual inserida.

O próximo estado partindo de INICIAL - através do acionamento do botão on, flag chamada push_on - é o INSERIRSENHA. Nele o display passa a exibir o valor selecionado através do potenciômetro (display=1) e o LED RGB passa a exibir a cor azul (led_b=1) indicando que o cofre está preparado para receber a senha atual. Partindo deste estado, existem três possibilidades:

- Continuar em INSERIRSENHA: através da combinação de duas flags - not(push_add) and not(push_on) - onde indicam que nem o botão on e nem o botão de adicionar foram pressionados. Ou seja, sem comandos externos o cofre irá aguardar para receber a senha;
- Ir para INICIAL: através do acionamento da flag push_on. Esta indica que o botão de ligar foi pressionado e com isso o cofre reinicia o processo indo para a configuração do estado INICIAL no qual está trancado;
- Ir para GRAVAR: através do acionamento da flag push_add. Esta indica que o botão de adicionar foi pressionado e com isso o cofre pode armazenar o valor de senha exibido pelo display, inserido por meio do potenciômetro.

Ao chegar em GRAVAR, o cofre irá armazenar a senha atual inserida e esperar neste estado por 0.5 segundos para então ir ao estado CONTADOR. As configurações de GRAVAR se constituem de apresentar o LED RGB em cor laranja, ativar o display para exibir o valor inserido (display=1), ativar o timer de 0.5 segundos e atribuir o sinal de entrada do

potenciômetro à um registrador. Para o LED, é necessária a combinação de intensidade das cores vermelha e verde para exibir laranja ($\text{led_r}=1$ e $\text{led_g}=0.5$). Para representar o armazenamento da senha inserida em fluxograma, utilizamos a variável reg_senhaN onde N representa o valor existente na reg_cont . Ou seja, caso o valor de reg_cont seja 2 a senha atual será inserida no reg_senha2 . Ainda neste estado, quando referente à atribuição do valor da senha atual, a seguinte equação foi escrita: $\text{reg_senha}[\text{reg_cont}] = \text{PC0}$. O termo PC0 se refere à porta C0 de entrada de sinal analógico à qual, como sugestão, o potenciômetro deverá ser conectado. A relação de portas a serem utilizadas por cada componente será tratado mais a frente.

Atingida a marca de 0.5 segundo no timer, o estado subsequente é o CONTADOR. Nele as configurações de display e LED se mantêm e além disso, é realizada a atualização da variável reg_cont . Esta variável tem como função principal contabilizar quantas senhas foram inseridas, por isso sua atualização é feita logo após à gravação do novo valor. Em seguida, para que exista o controle do loop e a quantidade exata de senhas seja inserida, um estado de comparação sugere a utilização da instrução CMP para verificar se o valor de reg_cont é igual a 2 e assim realizar outra inserção de senha ou partir para a verificação destas.

Neste estado de comparação interna do contador de senhas, a configuração apresentada no cofre é a mesma do estado GRAVAR. Existem três possíveis movimentos a partir deste estado de comparação:

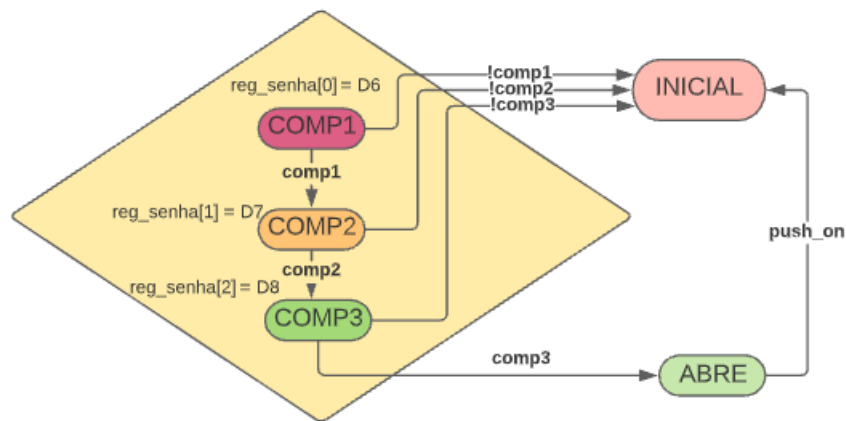
- (a) Se manter no estado: caso a contagem do timer de 0.5 segundos ainda não esteja finalizada - not (timer05) ;
- (b) Iniciar um novo ciclo: caso o valor existente em reg_cont seja menor que 3 e a contagem do timer de 0.5 segundo tenha sido encerrada;
- (c) Ir para a verificação de senhas: caso reg_cont tenha atingido o valor 3 e o timer tenha atingido o valor exigido de meio segundo.

O penúltimo estado, acessível por meio do estado de controle de loop, é o de verificação de senhas denominado "COMPARA SENHA". Com o propósito de analisar se os valores inseridos pelo usuário do cofre são os corretos, o estado possui as mesmas configurações de LED e display que GRAVAR e a partir dele pode ocorrer ou não a abertura do cofre.

- (a) Abertura do cofre: Caso a sequência de verificação realizada dentro do bloco atual retorne o valor esperado (os três valores corretos), o sistema segue para o estado ABRE;
- (b) Não abertura do cofre: Caso a verificação não retorne o esperado, alguns dos três valores inseridos estava incorreto, o sistema retorna ao estado INICIAL reiniciando todo o processo.

Para este bloco em específico foi desenvolvido uma lógica de verificação e para efeito de demonstração desta, foi criado um fluxograma próprio apresentado na Figura 8. Neste diagrama existe a possibilidade da realização de 1 até 3 verificações de valores. A lógica utilizada consiste da seguinte sequência: O primeiro valor armazenado é verificado e se este for igual ao valor de senha esperado, então o segundo valor inserido será verificado e assim por subsequente. Caso não, as verificações são encerradas e o sistema segue para o estado INICIAL sem abrir o cofre e isto vale para todas as verificações.

Figura 8: Estado Compara Senha

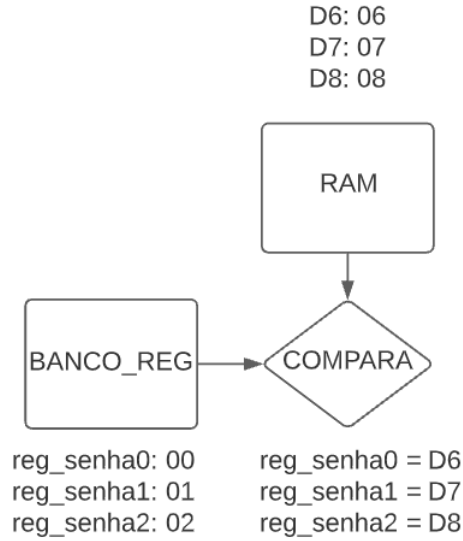


Fonte: Elaboração dos Autores.

A quesito de melhor explicação foram adicionadas linhas do pseudo algoritmo ao diagrama representando a comparação de um exemplo.

Neste exemplo, Figura 9, a memória RAM possui armazenado 3 valores de referência para comparação - a senha verdadeira - em três endereços diferentes (D6, D7, D8) e o banco de registradores armazenou os 3 valores inseridos para verificação, também em três endereços diferentes (reg_senha0, reg_senha1, reg_senha2). No estado COMPARA SENHA é realizada a comparação de um (caso o primeiro valor esteja errado), dois (caso o segundo valor esteja errado) ou três valores (caso todos estejam corretos ou apenas o último errado) para identificar se será permitida ou não a abertura do cofre. É este o exemplo descrito na figura 8 dentro do bloco de comparar senha.

Figura 9: Exemplo de Comparação



Fonte: Elaboração dos Autores.

O último estado a ser definido é o que realiza a abertura do cofre. Este foi denominado de ABRE e as configurações apresentadas nele são: o LED RGB apresentando a cor verde ($\text{led_g}=1$), o display acionado ($\text{display}=1$) apresentando 3 traços horizontais e enfim a abertura do cofre indicada por uma flag ($\text{aberto}=1$). Sendo este o último estado do total 7, dentro de um ciclo de abertura correta do sistema do cofre digital, as duas únicas opções de movimentação são: permanecer no estado, confirmado pelo não acionamento do botão on e, caso seja acionado, ir ao estado INICIAL. Fazendo assim, o trancamento do cofre digital e reiniciando todo o processo.

3.2 DETERMINAÇÃO DOS PINOS

A partir dos conhecimentos de Aloï (2015), os registradores de porta também chamados de “PORT”, possibilitam acesso mais eficiente das portas a partir das entradas e saídas do microcontrolador, em um placa Arduino. Em se tratando do ATMEGA328P, este possui *chips* em 3 canais de portas, são eles:

- (a) PORT B e D são pinos digitais;
- (b) PORT C são entradas analógicas.

Segundo Atmel (2016), cada pino de porta é composto por três *bits* locais de endereçamento de memória, são eles: o registrador de dados (PortX_n), dados de registros de direção (DDR_x) e os pinos de entrada da porta (PIN_x).

- (a) PortX_n : O registrador verifica se a porta está com nível lógico alto ou baixo;

- (b) DDR_x : O registrador verifica o sentido da porta se trata-se de uma saída ou de entrada;
- (c) PIN_x : o registrador efetua a leitura das portas de entradas.

Assim, se $PortX_n=1$ ele é configurado como pino de entrada e o resistor trata-se do tipo *pull-up*. Por outro lado, quando for feita uma escrita ($PortX_n=0$) e o pino configuração como saída terá nível lógico em “0” indo para *pull-down*. No DDR_x , estes registradores verificam se os pinos para aquela dada porta específica está como sendo entrada ou saída. Para isso, verifica se $DDR_x=1$ ele é dito como sendo saída. Por outro lado, para usar uma dada porta como sendo entrada $DDR_x=0$. Já PIN_x são registros somente de leitura. O valor do bit indica o nível lógico naquele pino. Se o pino tiver nível lógico alto, ele será lido. Caso contrário, não haverá leitura naquele dado instante. É bom lembrar, que após o sinal de *reset* DDR_x é zerado especificando-o como sendo entrada. Em virtude de facilitar o entendimento por parte do leitor, o sufixo “x” é substituído pela porta que está sendo utilizada, nesse caso $DDRB$, $DDRC$ e $DDRD$. Analogamente, as portas B,C e D são representadas por $PORTB$, $PORTC$ e $PORTD$.

De acordo com a Figura 10, por exemplo, $PORTB$ trata-se de uma porta que possui 8 *bits*. Nela temos a indicação de qual porta B está sendo utilizada ($PORTB7$ até $PORTB0$), se a informação contida nela está sendo lida ou escrita e o seu valor inicial.

Figura 10: $PORTB$ - Registro de dados

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fonte: Atmel (2016).

Em relação aos pinos referente ao $PORTB$, cada um possui uma função alternativa conforme pode ser visualizado na Figura 11.

Figura 11: Funções alternativas referente aos pinos do PORTB

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	\overline{SS} (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

Fonte: Atmel (2016).

Em relação aos pinos referente ao PORTC, cada um possui uma função alternativa conforme pode ser visualizado na Figura 12.

Figura 12: Funções alternativas referente aos pinos do PORTC

Port Pin	Alternate Function
PC6	RESET (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

Fonte: Atmel (2016).

Em relação aos pinos referente ao PORTD, cada um possui uma função alternativa conforme pode ser visualizado na Figura 13.

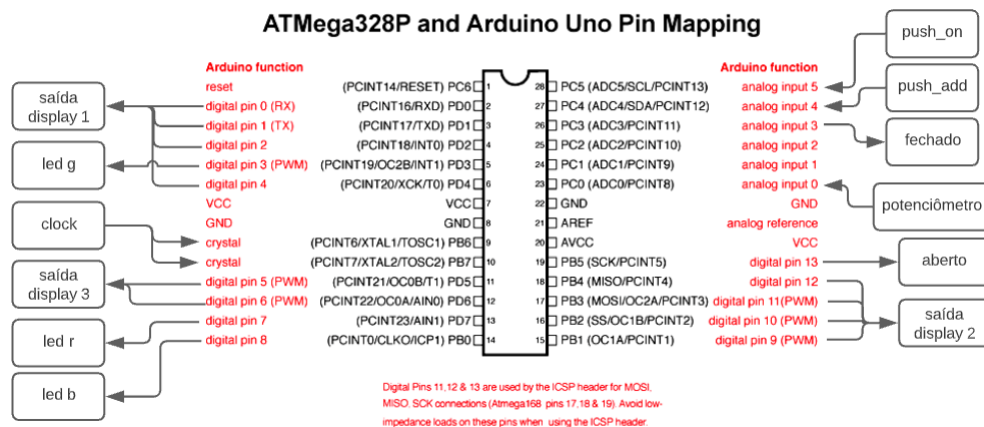
Figura 13: Funções alternativas referente aos pinos do PORTD

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

Fonte: Atmel (2016).

Para o referido projeto em questão, foi feita a conexão dos pinos do ATMEGA321 com os periféricos sugeridos no projeto ilustrado na Figura 14.

Figura 14: Mapa de pinos do ATMEGA328P



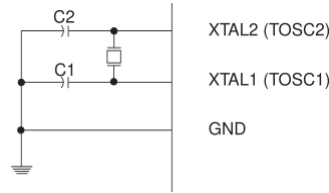
Fonte: Atmel (2016) <Adaptado>.

Os pinos do ATMEGA328P e os periféricos que os interligam mostrados na Figura 14 foram escolhidos da seguinte forma:

- (a) PB1 à PB4: Temos a saída que irá para o BIN-BCD para fazer o acionamento dos *displays* das dezenas;
- (b) PC0: Entrada analógica referente ao valor que aparecerá no display, conforme a variação da resistência do potenciômetro;
- (c) PC4 e PC5: Entrada analógica referente aos *pushbuttons* quando o usuário apertar o botão ON ou usuário inserir um novo valor para tentar inserir um novo valor para tentar acertar a senha;
- (d) PB5: Trata-se de uma *flag* de saída quando o cofre for para a condição do cofre estado aberto;
- (e) PC3: Trata-se de uma *flag* de saída quando o cofre for para a condição do cofre está fechado;
- (f) PB0: Saída de indicação quando o led azul for ligado e o comando do led vermelho e do led verde desligado do led RGB;
- (g) PD7: Saída de indicação quando o led vermelho for ligado e o led vermelho e o led verde desligado do led RGB;
- (h) PD3: Saída de indicação quando o led green for ligado e o led vermelho e o led blue desligados do led RGB;
- (i) PD5 e PD6: Temos a saída que irá para o BIN-BCD para fazer o acionamento dos *displays* das centenas;
- (j) PD0 a PD4: Temos a saída que irá para o BIN-BCD para fazer o acionamento dos *displays* das unidades;
- (k) PB6 e PB7: Duas entradas referente ao *clock* que estão relacionados aos capacitores entre o cristal.

Em relação ao item (k), as entradas PB6 e PB7 e suas conexões são representadas internamente a partir da Figura 15. Nela, temos o cristal entre os capacitores C1(PB6) e C2(PB7).

Figura 15: Entradas PB6 e PB7



Fonte: Atmel (2016).

O *clock* sugerido pelo grupo é de 16 MHz. Além disso, temos que os valores dos capacitores devem ser iguais ($C1=C2$) e devem estar entre 12 à 22 pF (pico-Faraday). No próximo tópico será descrito a conversão A/D.

3.3 CONVERSÃO A/D

As senhas serão inseridas de forma analógica através de um potenciômetro, para isto será necessária a utilização do conversor analógico-digital presente no ATmega328P, este conversor possui uma resolução de 10 bits, podendo assumir 1024 valores, ideal para este projeto, dado que os valores das senhas devem estar contidos no intervalo de 0 a 999, visto que serão utilizados 3 *displays* de 7 segmento para a exibição das mesmas.

Para a utilização do conversor analógico-digital temos que definir os parâmetros de quatro registradores sendo estes o *ADC multiplexer select* (ADMUX) e o *ADC ctrl & status register A* e B (ADCSRA e ADCSRB), *digital input disable register 0* (DIDR0) e para a leitura da informação convertida deve-se ler os registradores *ADC data register* (ADCH/ADCL), dois registradores de 8 bits que juntos compõem o registrador responsável por armazenar os 10 bits da mensagem convertida, estes registradores e as escolhas de projeto para seus parâmetros serão descritos a seguir.

O registrador ADMUX possui oito bits e segue a descrição da Figura 16, ele é o responsável pela definição do formato de exibição da leitura além das saídas de dois multiplexadores, os dois primeiros bits são responsáveis por definir a tensão de referência a ser utilizada pelo conversor, entre as opções disponíveis neste projeto definimos a tensão de referência como Vcc, a qual tem como bits seletores 01.

Figura 16: Registrador ADMUX

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fonte: Atmel.

O segundo é o multiplexador que define qual porta de entrada que será utilizada, entre as oito portas da família ADC, um sensor de temperaturas interno e as tensões de 1.1V (V_{BG}) e 0V (GND), como o nosso objetivo é a leitura de um potenciômetro será usada uma das portas da família ADC tendo os bits seletores definidos na Figura 17, onde neste projeto definimos como sendo a ADC0, porém pode ser utilizada qualquer porta ADC para esta função. Além desses bits o registrador possui o bit 5 responsável por definir o formato de escrita da conversão entre ajustado a direita ou à esquerda, e temos como escolha de projeto a definição da escrita ajustado à direita o que acarreta a esse bit ser definido com valor lógico 0, e o bit 4 o qual não pode ser alterado pois possui uso reservado.

Figura 17: Bits seletores de referência

MUX3...0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8 ⁽¹⁾
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V _{BG})
1111	0V (GND)

Fonte: Atmel (2016).

O registrador ADCSRA possui 8 bits distribuídos como descrito na Figura 18, o qual é responsável por realizar o controle do conversor, a seguir definiremos o estado destes seguindo o nosso projeto. ADEN corresponde ao habilitador do conversor, em nosso projeto definimos que o conversor estará ativado no momento de leitura de senha e será desativado em todos os outros momentos, para ativar o conversor este bit deve ser definido como 1. ADSC é responsável por iniciar a primeira conversão no modo de leitura o qual selecionamos, que é o de leituras contínuas (*free running*), sendo definido como 1 em todas as vezes que entramos na leitura de senhas. ADATE é responsável pelo ativamento automático do conversor, função esta que não utilizaremos, sendo assim

ele continuará definido como zero. ADIF e ADIE também não serão manipulados pois estes são utilizados quando existem no projeto rotinas de interrupções, o que não é o caso. ADPS[2:0] são responsáveis pela definição do fator de divisão do *clock*, como neste projeto o fator de divisão escolhido foi 2, estes bits serão definidos como 000.

Figura 18: Registrador ADCSRA

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fonte: Atmel (2016).

ADCSRB é o registrador onde definiremos o tipo de acionamento do conversor, descrito pela Figura 19. Neste registrador não será feita nenhuma mudança no bit ACME, pois este é usado apenas quando estão sendo feitas comparações de sinais analógicos, nem em ASTS[2:0] que são os bits que definem a forma de acionamento, pois como iremos utilizar o modo *free running* este será definido como 000, onde consequentemente não será necessário fazer nenhuma alteração pois estes já são os valores iniciais dos mesmos.

Figura 19: Registrador ADCSRB

Bit	7	6	5	4	3	2	1	0	
(0x7B)	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fonte: Atmel (2016).

DIDR0 é o registrador responsável por desabilitar o *buffer* de entrada digital do pino ADC, sendo assim, deve-se ser definido com valor lógico 1 o bit referente a porta escolhida para a entrada de informações do potenciômetro.

Os registradores ADCH/ADCL são apenas de leitura e é onde serão encontrados os resultados da conversão, ajustados a direita, como definido anteriormente, da forma ilustrada na Figura 20.

Figura 20: Registradores ADCH/ADCL

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Fonte: Atmel (2016).

3.4 Regulagem dos valores a partir do potencímetro

Para impedir que o usuário coloque valores acima de 999 no cofre, visto que os *displays* de 7 segmentos não podem exibir um valor maior que esse, fez-se necessário criar uma lógica utilizando uma fórmula matemática, afim de limitar os valores inseridos com o manuseio do potencímetro por parte do usuário. Isso foi feito, conforme a Fórmula 1. Vale ressaltar, “L” é o valor que do resultado da operação, “pot” representa o limite dos valores que o potencímetro pode mostrar (0 à 999), “fi_min” trata-se do valor mínimo na faixa de entrada, “fo_max” trata-se do valor máximo na faixa de saída, “fo_min” refere-se a faixa de saída mínima e “fi_max” é a faixa de entrada máxima.

Fórmula 1 - Limite dos valores a partir do potencímetro

$$L = (pot - fi_min) * (fo_max - fo_min) / (fi_max - (fi_min) + (fo_min)) \quad (1)$$

Fonte: Campos (2018).

No entanto, aconselha-se que os valores de “fi_min” e fo_min sejam “0” e os valores de fi_max=“1023” e fo_max= “999”. Isso porque, é a faixa dos valores máximos dos 10 bits “fi_max” e “fo_max” é a faixa máxima do potencímetro. Dessa forma, os valores serão mostrados corretamente no *display*, não ultrapassando o valor 999, nem exibindo valores negativos.

Fórmula 2 - Limite dos valores a partir do potencímetro, com valores substituídos

$$L = (pot * 999) / 1023 \quad (2)$$

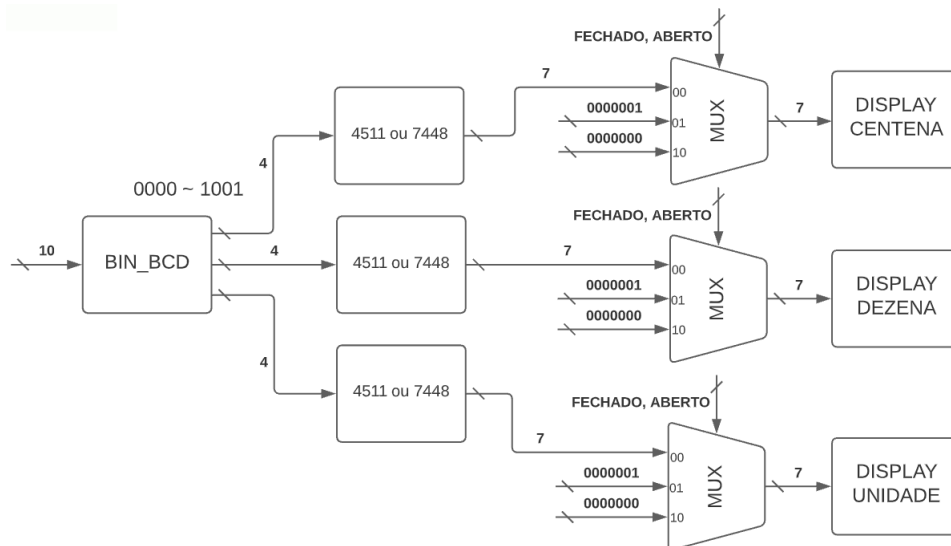
Fonte: Campos (2018) <adaptado>.

3.5 DISPLAY E LEDS

A partir da conversão analógica-digital realizada, têm-se um valor binário de barramento igual a 10bits, este valor que será atualizado a cada mudança de resistência do potenciômetro deve ser lido nos pinos digitais do microcontrolador ATmega328P.

Esse dado de 10 bits lido entrará em um circuito conversor binário para *Binary-Coded Decimal* (BCD), onde irá gerar três outras saídas de barramento 4 bits cada, tais saídas serão correspondentes ao BCD da unidade, dezena e centena do valor convertido. Como estes valores terão de aparecer em *displays* de 7 segmentos, esses dados de 4 bits terão de entrar em um CI conversor BCD para código de 7 segmentos, ou seja, irá converter o valor de 4 bits da unidade, dezena e centena em outro valor de barramento de 7 bits referentes a cada segmento dos *displays* de 7 segmentos. Na Figura 21 é possível visualizar o diagrama de blocos da ideia pensada para os displays.

Figura 21: Diagrama de blocos da conversão para os displays



Fonte: Elaborado pelos autores.

Levando em conta que o cofre quando está em aberto possui uma configuração em que todos os displays irão acender apenas o segmento do meio e também a configuração do estado inicial em que não irá acender nenhum segmento do display, foi pensado em uma solução que irá funcionar através de duas *flags* que irão sair através de pinos setados para saírem valores digitais do microcontrolador denominada de "*FECHADO*" e "*ABERTO*" e servirão de chave seletora de 21 multiplexadores 4x1, cada multiplexador selecionará bit a bit da saída que irá acender os segmentos do display, a Tabela 1 mostra a relação das entradas seletoras dos multiplexadores com as saídas esperadas.

Tabela 1: Chave seletora e saída dos multiplexadores

FECHADO / ABERTO	SAÍDA DO MUX
00	SAÍDA DO 4511 ou 7448
01	0000001
10	0000000

Fonte: Elaborado pelos autores.

O LED do painel servirá para indicar ao usuário a condição que o cofre está no momento. Quando LED estiver vermelho, indica que o cofre está no estado inicial e o cofre está fechado, o LED azul indica ao usuário que está esperando uma senha ser adicionada e o cofre ainda permanece fechado, o LED laranja indica que alguma operação está acontecendo no cofre, seja de armazenamento da senha inserida, ou contando quantas senhas já foram inseridas para abrir ou não o cofre, por fim, o LED verde indica que o cofre está aberto. A relação entre os estados e a saída em que os pinos de cada terminal do led RGB está na Tabela 2.

Tabela 2: *Duty cycles* para cada pino dos terminais do LED RGB

ESTADO	Duty Cycle		
	pin_LED R	pin_LED G	pin_LED B
Inicial	100%	0%	0%
InserirSenha	0%	0%	100%
Gravar	100%	50%	0%
Abre	0%	100%	0%

Fonte: Elaborado pelos autores

Perceba que o pino referente ao terminal do verde as vezes deve receber uma onda quadrada de 50% de *duty cycle*, isso porque para criar a cor laranja no LED RGB é necessário a cor vermelha misturada com metade do que seria a cor verde, isso é alcançado através de um pino digital com função PWM, pois ele alterna entre o valor 0 e 1 em uma frequência suficiente para que o LED fique laranja invés de amarelo.

3.6 TEMPORIZADOR DE 0,5 SEGUNDOS

O sistema opera em um *clock* na casa dos milhões (10^6), portanto, para definir o temporizador é preciso ser feita uma manipulação do *clock* usado. Para tal alteração é necessário utilizar o *clock* do ATMEGA328P, o qual opera na frequência de 8MHz. Na tabela mostrada na Figura 22, são apresentadas as divisões que podem ser realizadas com o *clock* interno.

Figura 22: Tabela de divisões do *clock* interno

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$\text{clk}_{\text{IO}}/(\text{No prescaling})$
0	1	0	$\text{clk}_{\text{IO}}/8$ (From prescaler)
0	1	1	$\text{clk}_{\text{IO}}/64$ (From prescaler)
1	0	0	$\text{clk}_{\text{IO}}/256$ (From prescaler)
1	0	1	$\text{clk}_{\text{IO}}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Fonte: Atmel (2016).

Por decisão de projeto, optamos por sugerir que seja feita uma divisão tal qual a mostrada na quinta linha da tabela, ou seja, uma divisão do *clock* interno por 1024. O resultado dessa divisão fará com que o *timer* opere por 1 segundo, como o temporizador projetado é de meio segundo é feita uma divisão por 2. O resultado encontrado é 3906, por essa razão recomenda-se o uso do *Timer* 1 do ATMEGA328P.

Referências

- [1] ALOI, RENATO. **Registradores de Portas**. Disponível em: <<http://renatoaloi.blogspot.com/2015/09/arduino-manipulacao-direta-de-portas.html>>. Acesso em 08/07/2021.
- [2] ATMEL, CORPORATION. **Atmel AVR 8-bit Instruction Set**, 2016. Disponível em: <<https://datasheetspdf.com/pdf-file/1469778/ATMEL/ATmega328P/1>>. Acesso em 06/07/2021.
- [3] CAMPOS, BRUNO. **Como funciona a função MAP do Arduino**, 2018. Disponível em: <<https://www.youtube.com/watch?v=OTATFv68Koo>>. Acesso em 11/07/2021.
- [4] DIAS, SAMAHERNI MORAIS. **Dados do problema 03 - Sistemas Digitais**. Natal, 2021.
- [5] GOPAKUMAR, AMITH. **Output Devices**. Disponível em: <<http://fabacademy.org/2018/labs/fablabtrivandrum/students/amith-gnair/week12.html>>. Acesso em 11 jul 2021.
- [6] PÉREZ, ALFONSO. **Conhecendo o Cerne do Microcontrolador Atmega328p Arduino Uno. (MIC165)**. 2018. Disponível em: <<https://www.newtoncbraga.com.br/index.php/microcontroladores/138-atmel/14863-conhecendo-o-cerne-do-microcontrolador-atmega328p-arduino-uno-mic165.html>>. Acesso em 11 jul 2021.
- [7] THOMSEN, ADILSON. **O que é Arduino?** 2014. Disponível em: <[https://www.filipeflop.com/blog/o-que-e-arduino/: :text=O%20Arduino%20foi%20criado%20em,a%20estudantes%20e%20projetistas%20a](https://www.filipeflop.com/blog/o-que-e-arduino/:%3Atext=O%20Arduino%20foi%20criado%20em,a%20estudantes%20e%20projetistas%20a)>. Acesso em 11 jul 2021.

ANEXOS

ANEXO A - Relato semanal

Líder: Isaac de Lyra Junior

A.1 Equipe

Tabela 3: Identificação da equipe

Função no grupo	Nome completo do aluno
Redator	Marcelo Ferreira Mota Júnior
Debatedor	Sthefania Fernandes Silva
Videomaker	Allysson de Andrade Silva
Auxiliar	Anny Beatriz Pinheiro Fernandes

Fonte: Produzido pelos autores.

A.2 Defina o problema

A problemática desta semana consiste em desenvolver o projeto de um circuito que retrate o funcionamento de um cofre digital. E com isso, na semana seguinte, seja possível a implementação deste sistema utilizando do microcontrolador ATmega328p para executar um código, em linguagem *assembly*, que controla a abertura da porta do cofre. Este cofre possui uma interface homem-máquina baseada em dois botões (On e Adicionar), um LED RGB, um potenciômetro e um *display* de 3 dígitos. Através destes componentes é possível realizar a inserção dos 3 valores necessários de senha e também verificar o estado atual do cofre (fechado, inserção de senha, verificação de senha, aberto).

Em conhecimento destas configurações, o projeto consiste em realizar um diagrama de blocos que retrate com eficiência o funcionamento e etapas de abertura do cofre. Além do estudo aprofundado a respeito do componente ATmega328p e seus periféricos, necessários em conhecimento para a implementação posterior do código fonte.

A.3 Registro de *brainstorming*

A primeira reunião foi bem breve e ocorreu no dia 06/07/2021, logo após a aula da disciplina, tal reunião foi utilizada para definição dos cargos e horários que utilizaremos para os encontros durante a semana, foi marcado a próxima reunião para o dia 08/07/2021, isso foi decidido para que todos pudessem estudar a arquitetura do microcontrolador ATmega328P e pensar nos blocos que seriam utilizados no projeto.

A segunda reunião aconteceu no dia 08/07/2021, foi iniciado a discussão sobre o diagrama de blocos do comportamento esperado pelo cofre, ainda nessa reunião foi discutido

sobre os blocos necessários da arquitetura do microcontrolador para que tal comportamento fosse atendido, ficou como atividade da próxima reunião estudar sobre os pinos de entrada e saída do ATmega328P e seus diferentes tipos.

A terceira reunião aconteceu no dia 09/07/2021, nela foi discutido sobre os estudos realizados acerca dos I/O *ports* e quais iríamos utilizar no projeto e também foi realizado alguns ajustes na MDE, ficou como atividade da próxima reunião estudar sobre o bloco conversor analógico-digital.

Na quarta reunião, realizada no dia 10/07/2021, foi discutido sobre os estudos realizados sobre o bloco conversor analógico-digital, onde foi discutido como armazenaríamos os 10 bits, já que os registradores possuem barramento de 8 bits e também sobre como a saída de 10 bits do valor convertido iria ser mostrada nos *displays*, nesta reunião foi decidido que utilizaríamos um conversor BIN-BCD, um conversor BCD-to-7S e um sistema de multiplexadores para as condições especiais do display. Ficou como atividade da próxima reunião estudar sobre o timer e o clock.

A quinta reunião(11/07/2021) foi utilizada para discutir os estudos realizados sobre o bloco de timer/counter, foi decidido que iríamos utilizar o bloco timer de barramento 16 bits da arquitetura do ATmega328P e que utilizaríamos um fator de divisão no clock que vai para este timer. Ainda nesta reunião foi dividido as partes que cada um ficaria responsável no relatório e iniciamos a escrita do mesmo.

A última reunião(12/07/2021) foi destinada para terminar a escrita do relatório e discutir possíveis ajustes necessários no projeto.

A.4 Pontos-chaves

Os principais aspectos a serem pontuados neste projeto são relacionados à conexão dos componentes externos ao microcontrolador, a utilização de portas analógicas e digitais, o controle do clock e da conversão analógica-digital e, também, a utilização e controle dos registradores internos do ATmega328p. Além do total entendimento de cada etapa de funcionalidades do cofre digital a ser implementado ao que se refere à abertura de sua porta.

A.5 Questões de pesquisa

Foram necessárias pesquisas quanto aos seguintes aspectos:

1. Pinagem, funcionamento e configurações básicas de conexão:
 - (a) do ATmega328p;
 - (b) de um potenciômetro;
 - (c) do LED RGB;

(d) de *push-buttons*.

2. Conversão Analógica-Digital do ATmega328p;
3. Clock do ATmega328p;
4. Utilização de PWM;
5. Controle e utilização de registradores internos do ATmega328p.

A.6 Planejamento da pesquisa

As reuniões ocorreram durante todos os dias da semana, com exceção da quarta, e em todos os dias discutíamos os tópicos a serem estudados para a próxima reunião, primeiramente estudamos sobre a arquitetura do microcontrolador ATmega328P, em seguida sobre os I/O *ports* e suas diferentes funções, o conversor A/D do microcontrolador e, por fim, foi estudado sobre o *timer* e *clock* do dispositivo e em que como utilizaríamos. Os horários das reuniões se encontra em verde na Figura 23.

Figura 23: Horários das reuniões

Horários	Terça	Quarta	Quinta	Sexta	Sábado	Domingo	Segunda
M12							
M34							
M56							
T12							
T34							
T56							
N12							
N34							

Fonte: Elaborado pelos autores