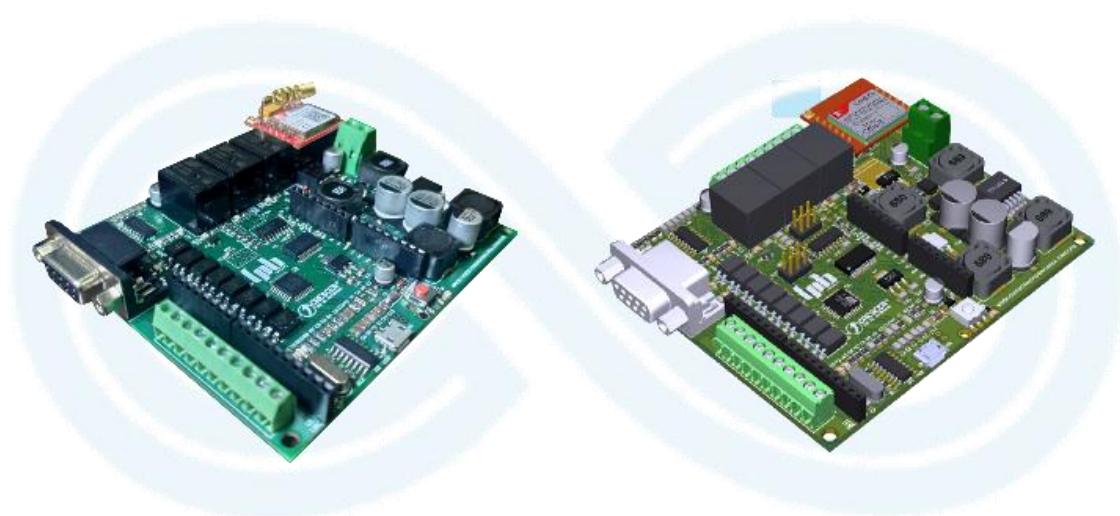




Crescer Professional Board

CPB INFINITY

Manual de Utilização





SUMÁRIO

1. INFORMAÇÕES GERAIS	4
2. SOBRE A CPB.....	4
3. HARDWARE	5
3.1 PINAGEM.....	6
3.2 APLICAÇÕES	7
4. SOFTWARE	8
5. ESPECIFICAÇÕES DAS FUNCIONALIDADES.....	9
5.1 FONTE DE ALIMENTAÇÃO	9
5.1.1 Hardware.....	9
5.2 COMUNICAÇÃO GSM	10
5.2.1 Hardware.....	10
5.2.2 Software	11
5.3 MULTIPLEXADOR.....	13
5.3.1 Hardware.....	13
5.3.2 Software	14
5.4 BARRAMENTO.....	15
5.4.1 Hardware.....	15
5.4.2 Software	16
5.5 SAÍDAS A RELÉS.....	17
5.5.1 Hardware.....	17
5.5.2 Software	18
5.6 COMUNICAÇÃO RS232 ou RS485	19
5.6.1 Hardware.....	19
5.6.2 Software	21
5.7 OPTOACOPLADORES.....	22
5.7.1 Hardware.....	22



5.7.2 Software	25
6. CONFIGURAÇÕES INFINITY	26
7. CONCLUSÃO	26



1. INFORMAÇÕES GERAIS

Este manual fornece informações necessárias para a correta utilização de todas as funções da Crescer Professional Board. O texto demonstra ao usuário, de forma objetiva, instruções exemplificadas para colocar em operações todas as funcionalidades da placa.

2. SOBRE A CPB

A CPB é uma solução em Hardware robusta e qualificada que viabiliza o emprego do Arduino, de maneira profissional, em qualquer produto ou equipamento independente da área de atuação.

Ideal para o desenvolvimento de soluções, seu objetivo é diminuir o tempo e dinheiro investido para viabilizar seu negócio ou produto, assumindo o viés técnico, a CPB permite direcionar os recursos dos usuários para o setor comercial alavancando suas vendas.

Por usar o conceito do arduino, a CPB tem integrações com inúmeras shields, sensores e atuadores de melhor custo-benefício, além de permitir que os usuários tenham uma base para a multiconectividade, podendo ser conectada a Nuvem com o seu módulo onboard de Comunicação GPRS/TCP IP ou mesmo com shields de conexão Wi-fi, Ethernet Cabeada, Bluetooth, LoRa.

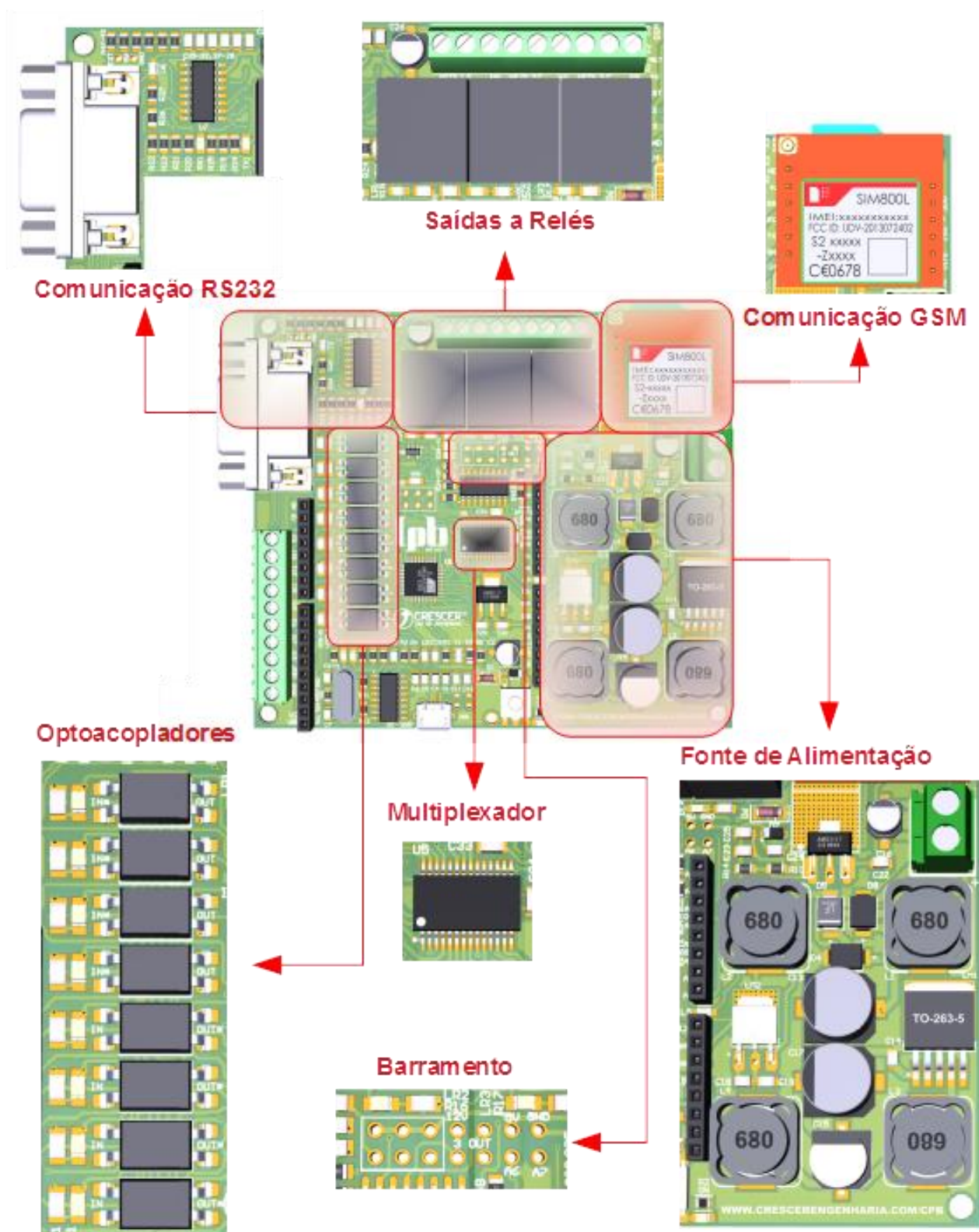
Vídeo de Lançamento: <https://www.youtube.com/watch?v=KgLikDG5SNE>

A CPB utiliza o Microcontrolador Atmega328p e Interface de Desenvolvimento (IDE) do Arduino para programação.

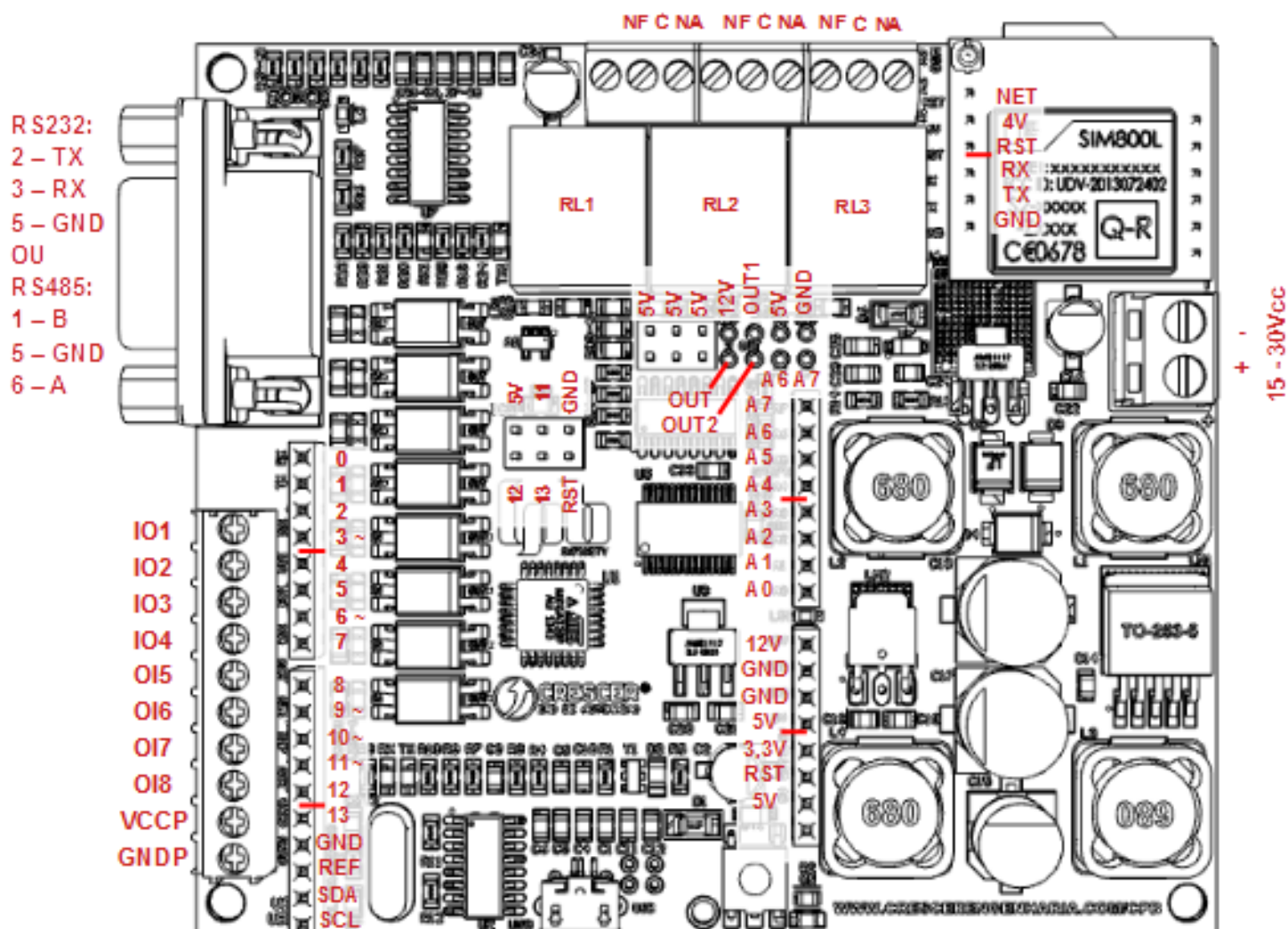
Datasheet Atmega328p: <https://datasheetspdf.com/pdf-file/1057332/ATMEL/ATmega328P/1>

3. HARDWARE

Este capítulo aborda uma mostra inicial do Hardware, separando-o em partes para uma posterior especificação.



3.1 PINAGEM



Pino	Legenda	Grupo
IO1-IO4	Entrada NPN	Optoacopladores
OI5-OI8	Saída Coletor Aberto	
0-13	Pinos digitais	Microcontrolador
~	Pinos digitais PWM	
REF, SDA e SCL	REF e barramento I2C	
A0-A7	Pinos Analógicos	
RST	Reset do Microcontrolador	
NF	Normalmente Fechado	Relés
NA	Normalmente Aberto	
C	Comum	
RL1, RL2, RL3	Relés	Externos
OUT, OUT1 e 2	Saídas Digitais para Relés Externos	
+, -	Alimentação	Comunicação GPRS/TCP IP
NET, 4V, RST, RX, TX E GND	Pinagem SIM800	
1, 2, 3, 5, 6	Pinagem DB9	Comunicação Serial



3.2 APLICAÇÕES

Produtos e negócios em que a CPB está empregada:

Cama de Quiropraxia;

Varredora de Escada Rolante;

Bobinadora de Filamento para impressão 3D;

Envasadora de Cianoacrilato;

Equipamentos de Eletromobilidade;

Equipamentos Calçadistas;

Equipamentos Agrônomos: estações meteorológicas e monitoramento de silos de grãos;

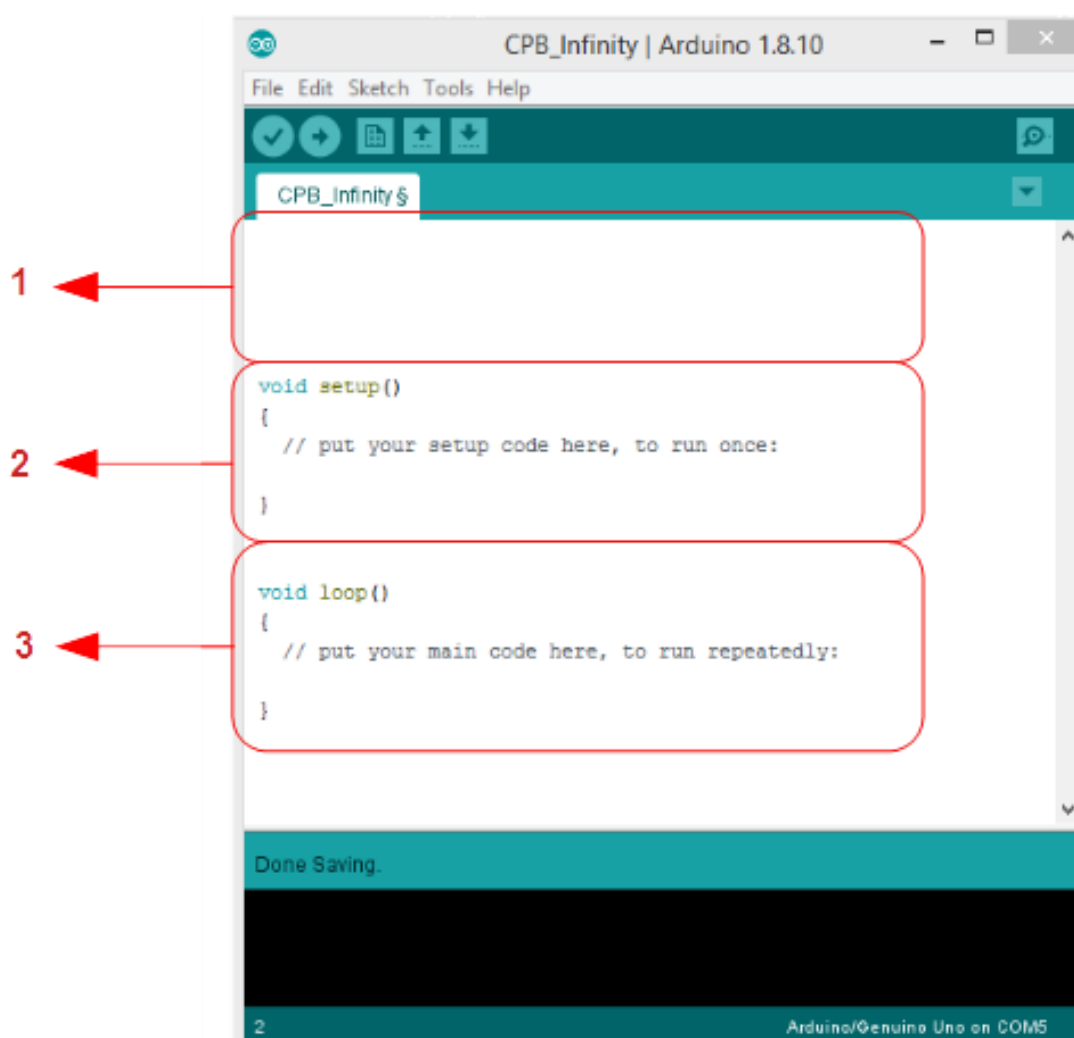
Equipamentos para aquisição de variáveis in loco para Softwares de TI.

4. SOFTWARE

Este capítulo aborda uma mostra inicial da interface de desenvolvimento do Software para a CPB, do qual é exigido um conhecimento básico de programação do Arduino.

Link de Indicação do curso: <https://www.udemy.com/course/automacao-profissional-com-arduino-completo/?referralCode=E663758FD3F7FD7A418F>

Link para recebimento do primeiro Módulo do Curso Gratuito: <https://www.crescerengenharia.com/cursogratis>



Legenda:

1. Região das variáveis e bibliotecas.
2. Setup.
3. Loop.

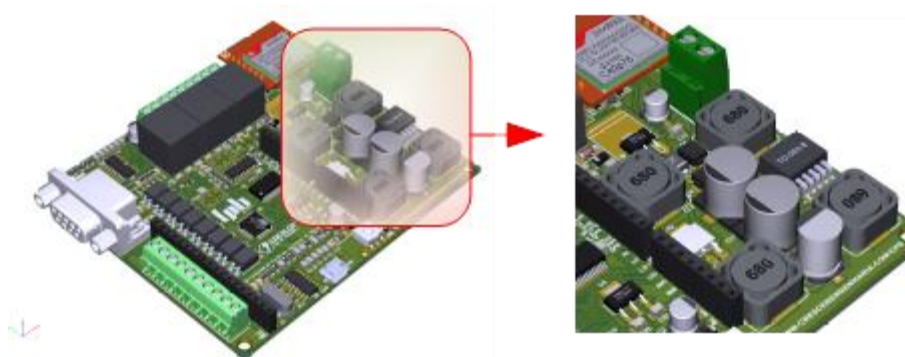
5. ESPECIFICAÇÕES DAS FUNCIONALIDADES

Os próximos capítulos especificaram as características de Hardware e Software de todas as funcionalidades anteriormente listadas.

5.1 FONTE DE ALIMENTAÇÃO

A CPB conta com uma fonte de alimentação robusta e testada para ser empregada em qualquer ambiente, inclusive setores Industriais.

5.1.1 Hardware



Borne para alimentação.

Tensão de Entrada: 15-30V.

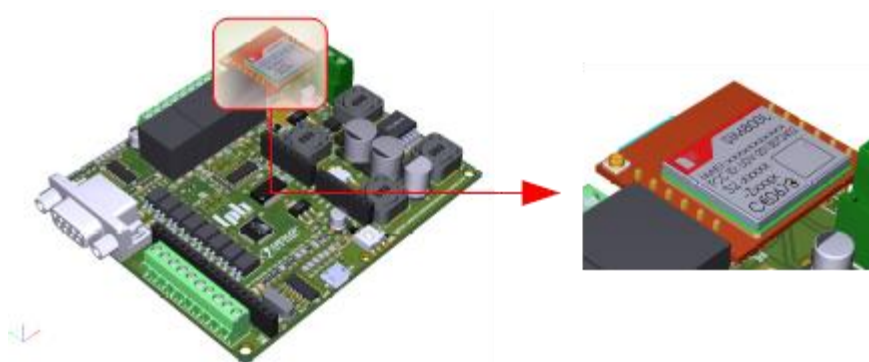
Consumo: 6 W.

Polaridade vide na placa.

5.2 COMUNICAÇÃO GSM

Um dos grandes diferenciais da CPB é a Comunicação GSM (GPRS e TCP/IP), da qual utiliza Chips de celular para enviar e receber SMS/ligações ou conectar-se na Web, com isso, permite a coleta e acionamento de dados e atuadores in loco para sistemas de TI, Telecom, Automação.

5.2.1 Hardware



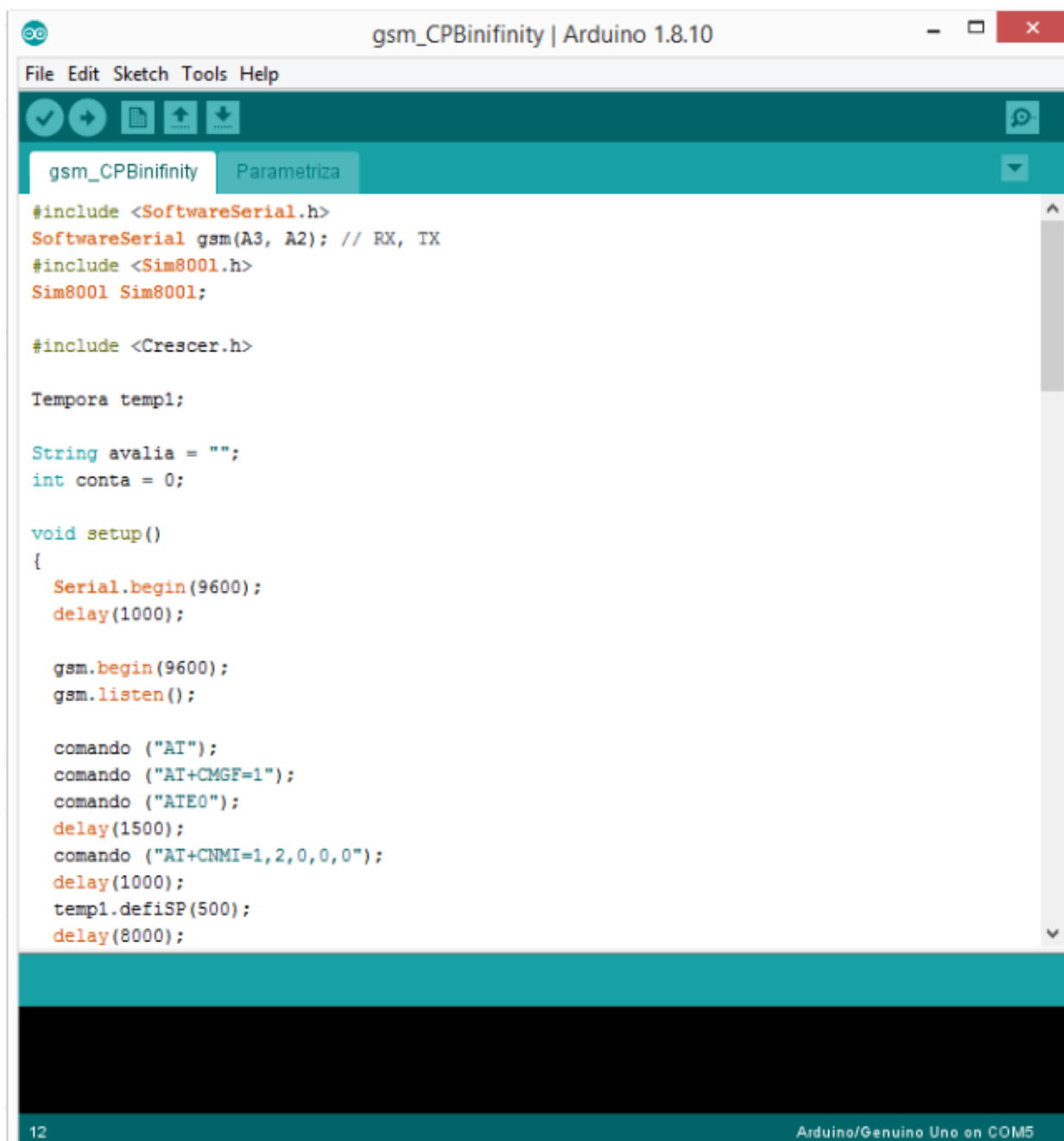
Shield GSM SIM800L.

Tensão de alimentação 4V - onboard.

Shield Plug and Play, basta programar CPB para a comunicação GSM de acordo com o exemplo a seguir.

5.2.2 Software

Exemplo de Software da CPB com Comunicação GSM:



```
gsm_CPBinfinity | Arduino 1.8.10
File Edit Sketch Tools Help
gsm_CPBinfinity Parametriza

#include <SoftwareSerial.h>
SoftwareSerial gsm(A3, A2); // RX, TX
#include <Sim8001.h>
Sim8001 Sim8001;

#include <Crescer.h>

Tempora temp1;

String avalia = "";
int conta = 0;

void setup()
{
  Serial.begin(9600);
  delay(1000);

  gsm.begin(9600);
  gsm.listen();

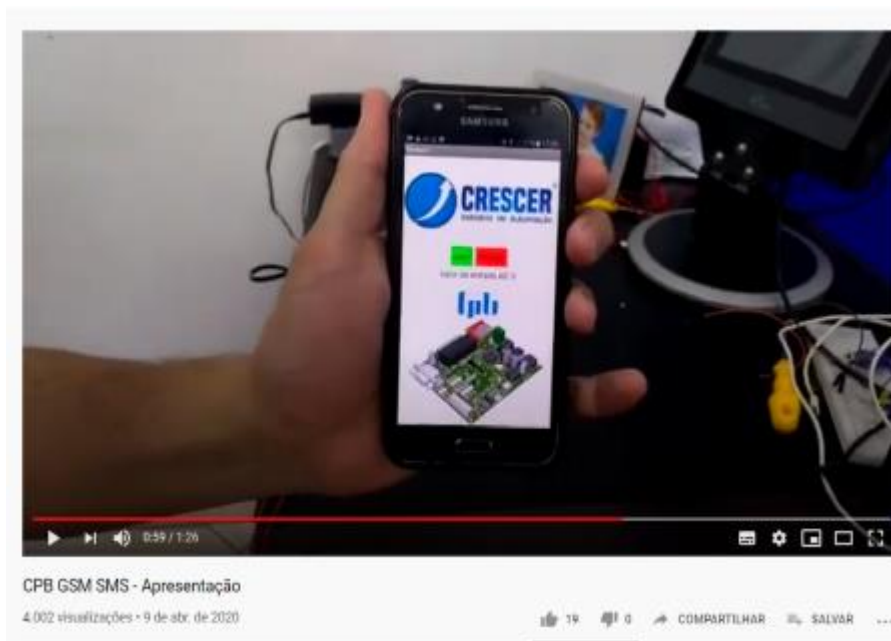
  comando ("AT");
  comando ("AT+CMGF=1");
  comando ("ATE0");
  delay(1500);
  comando ("AT+CNMI=1,2,0,0,0");
  delay(1000);
  temp1.defiSP(500);
  delay(8000);
}
```

Imagem somente do Setup devido a extensão do código, completo abaixo:

Exemplo "gsm_CPBinfinity", download disponível em:

<https://github.com/casaautomacao/CPBinfinity>

Exemplo do funcionamento com uma série de vídeos no youtube:



Apresentação:

Inicial - <https://www.youtube.com/watch?v=u4CRPSZ01y8&t=3s>

Explicação e Funcionamento:

Parte 1 - <https://www.youtube.com/watch?v=XJn5GEmCuqM>

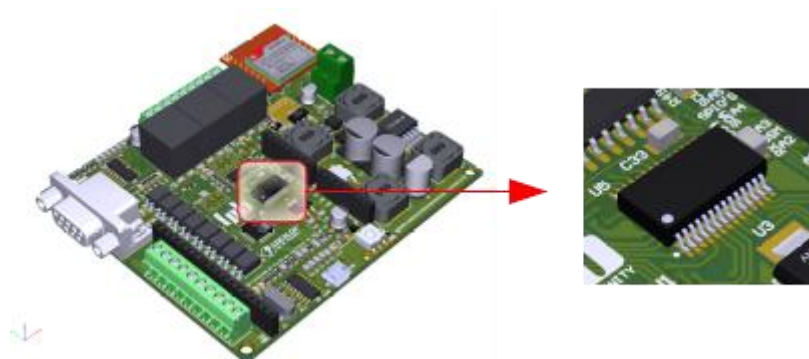
Parte 2 - https://www.youtube.com/watch?v=yXZ_SBmqF8w

Parte 3 - <https://www.youtube.com/watch?v=9zWuNoa4D-0>

5.3 MULTIPLEXADOR

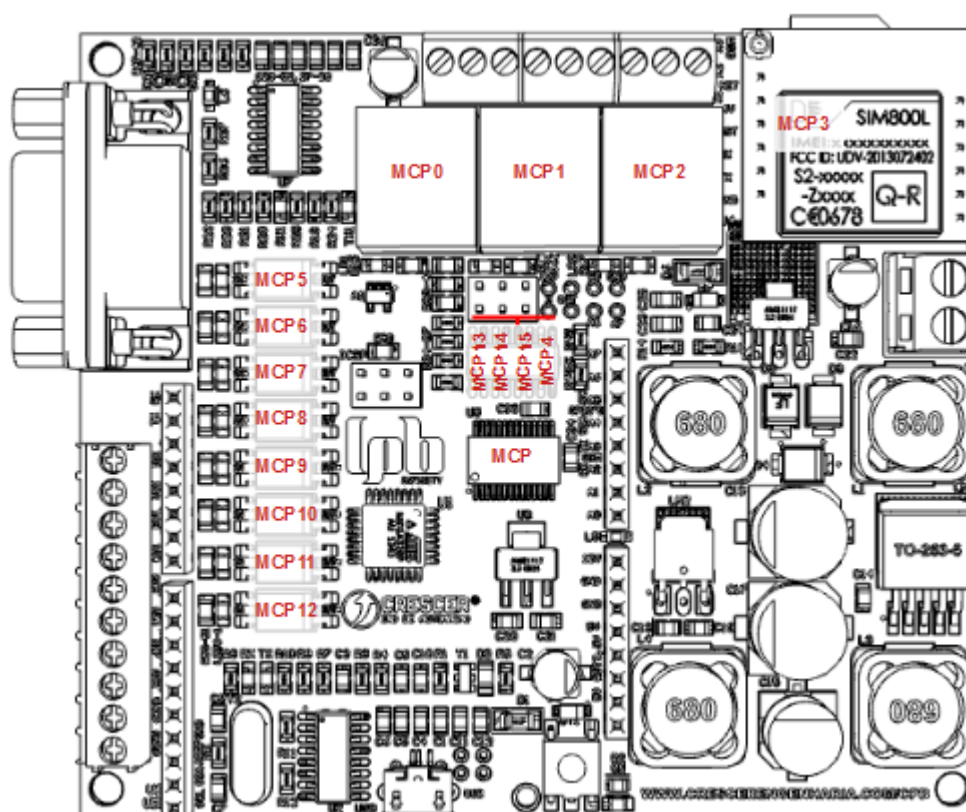
A CPB conta com um Chip Multiplexador para utilizar menos pinos do Microcontrolador e ter um aumento na quantidade disponíveis de I/O's.

5.3.1 Hardware



A multiplexação da CPB é feita pelo Chip MCP23017. Este multiplexador parametriza 16(0 ao 15) pinos como I/O e os controla via barramento I2C.

Datasheet: <http://ww1.microchip.com/downloads/en/devicedoc/20001952c.pdf>

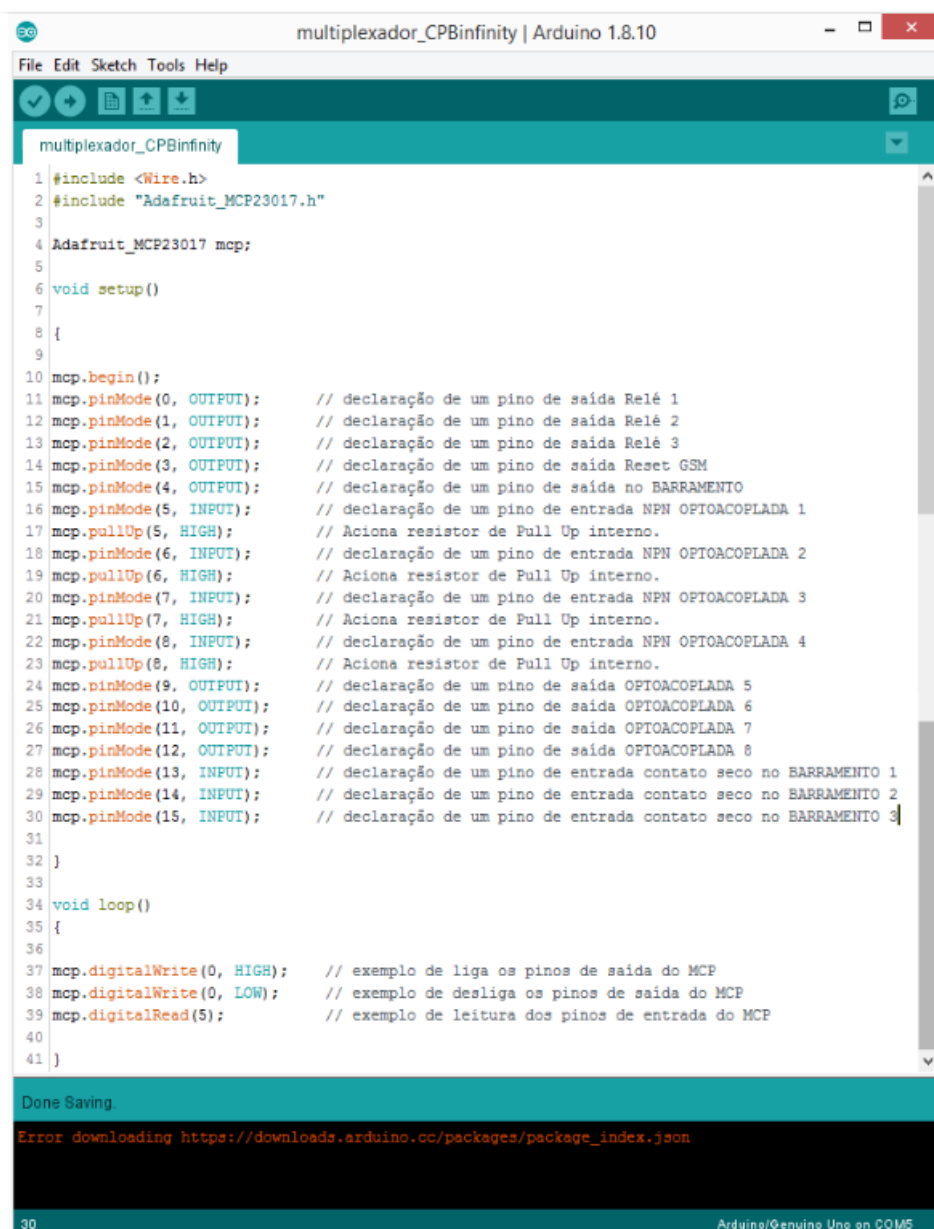


5.3.2 Software

Adicione ao seu Software a biblioteca do MCP23017 para que sua CPB tenha sua Pinagem configurável ativa.

Link para download da biblioteca: <https://github.com/adafruit/Adafruit-MCP23017-Arduino-Library>

Exemplo de Software do MCP23017 com aplicação prática dos pinos da CPB:



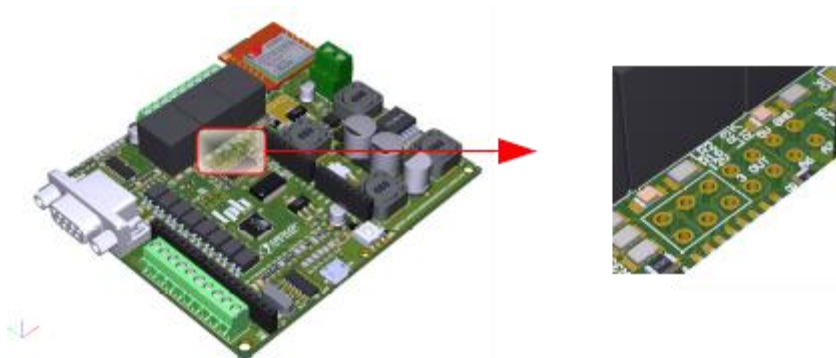
```
multiplexador_CPBinfinity | Arduino 1.8.10
File Edit Sketch Tools Help
multiplexador_CPBinfinity
1 #include <Wire.h>
2 #include "Adafruit_MCP23017.h"
3
4 Adafruit_MCP23017 mcp;
5
6 void setup()
7 {
8
9
10 mcp.begin();
11 mcp.pinMode(0, OUTPUT); // declaração de um pino de saída Relê 1
12 mcp.pinMode(1, OUTPUT); // declaração de um pino de saída Relê 2
13 mcp.pinMode(2, OUTPUT); // declaração de um pino de saída Relê 3
14 mcp.pinMode(3, OUTPUT); // declaração de um pino de saída Reset GSM
15 mcp.pinMode(4, OUTPUT); // declaração de um pino de saída no BARRAMENTO
16 mcp.pinMode(5, INPUT); // declaração de um pino de entrada NPN OPTOACOPLADA 1
17 mcp.pullUp(5, HIGH); // Aciona resistor de Pull Up interno.
18 mcp.pinMode(6, INPUT); // declaração de um pino de entrada NPN OPTOACOPLADA 2
19 mcp.pullUp(6, HIGH); // Aciona resistor de Pull Up interno.
20 mcp.pinMode(7, INPUT); // declaração de um pino de entrada NPN OPTOACOPLADA 3
21 mcp.pullUp(7, HIGH); // Aciona resistor de Pull Up interno.
22 mcp.pinMode(8, INPUT); // declaração de um pino de entrada NPN OPTOACOPLADA 4
23 mcp.pullUp(8, HIGH); // Aciona resistor de Pull Up interno.
24 mcp.pinMode(9, OUTPUT); // declaração de um pino de saída OPTOACOPLADA 5
25 mcp.pinMode(10, OUTPUT); // declaração de um pino de saída OPTOACOPLADA 6
26 mcp.pinMode(11, OUTPUT); // declaração de um pino de saída OPTOACOPLADA 7
27 mcp.pinMode(12, OUTPUT); // declaração de um pino de saída OPTOACOPLADA 8
28 mcp.pinMode(13, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO 1
29 mcp.pinMode(14, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO 2
30 mcp.pinMode(15, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO 3
31
32 }
33
34 void loop()
35 {
36
37 mcp.digitalWrite(0, HIGH); // exemplo de liga os pinos de saída do MCP
38 mcp.digitalWrite(0, LOW); // exemplo de desliga os pinos de saída do MCP
39 mcp.digitalRead(5); // exemplo de leitura dos pinos de entrada do MCP
40
41 }
Done Saving.
Error downloading https://downloads.arduino.cc/packages/package_index.json
30 Arduino/Genuino Uno on COM5
```

Exemplo “multiplexador_CPBinfinity”, download disponível em: <https://github.com/casaautomacao/CPBinfinity>

5.4 BARRAMENTO

O barramento CPB Infinity possui algumas pinagens auxiliares que facilitam no interfaceamento com shields e atuadores externos.

5.4.1 Hardware

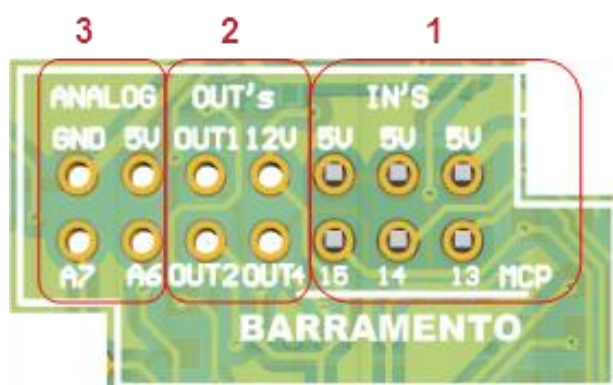


O barramento possui:

1 – 3 entradas digitais (com resistores de pulldown na placa, somente para aplicações de contato seco. Sendo obrigatório o uso do 5V disponível neste pino).

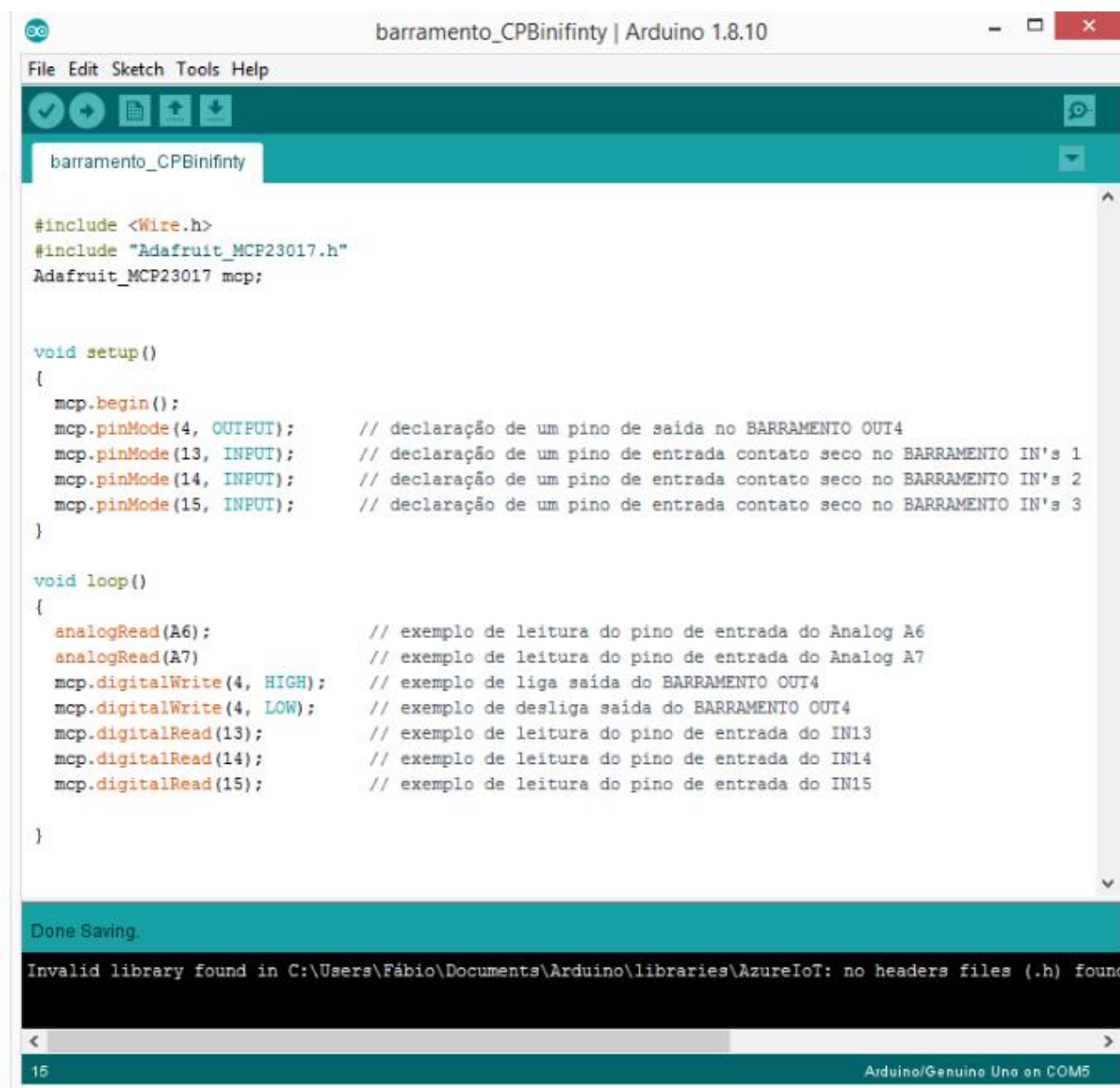
2 – 3 saídas digitais para relés externos.

3 – Slot para shields analógicas, que tem alimentação (5V e GND) e os pinos A6 e A7.



5.4.2 Software

Exemplo de Software com as entradas digitais da CPB:



```
#include <Wire.h>
#include "Adafruit_MCP23017.h"
Adafruit_MCP23017 mcp;

void setup()
{
  mcp.begin();
  mcp.pinMode(4, OUTPUT); // declaração de um pino de saída no BARRAMENTO OUT4
  mcp.pinMode(13, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO IN's 1
  mcp.pinMode(14, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO IN's 2
  mcp.pinMode(15, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO IN's 3
}

void loop()
{
  analogRead(A6); // exemplo de leitura do pino de entrada do Analog A6
  analogRead(A7); // exemplo de leitura do pino de entrada do Analog A7
  mcp.digitalWrite(4, HIGH); // exemplo de liga saída do BARRAMENTO OUT4
  mcp.digitalWrite(4, LOW); // exemplo de desliga saída do BARRAMENTO OUT4
  mcp.digitalRead(13); // exemplo de leitura do pino de entrada do IN13
  mcp.digitalRead(14); // exemplo de leitura do pino de entrada do IN14
  mcp.digitalRead(15); // exemplo de leitura do pino de entrada do IN15
}
```

Done Saving.

Invalid library found in C:\Users\Fábio\Documents\Arduino\libraries\AzureIoT: no headers files (.h) found

15 Arduino/Genuino Uno on COM5

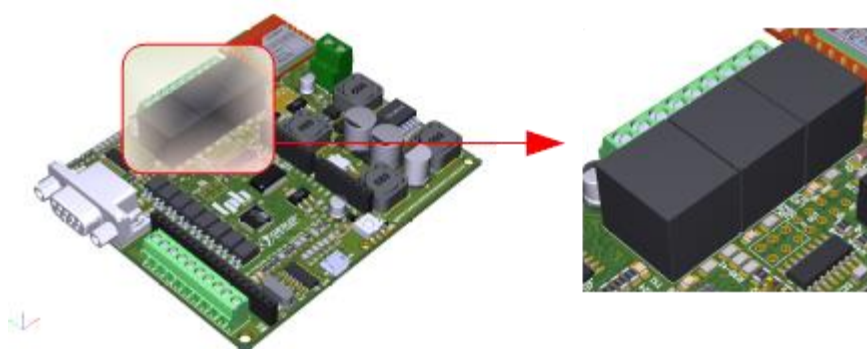
Exemplo “barramento_CPBinifinty”, download disponível em:

<https://github.com/casaautomacao/CPBinifinty>

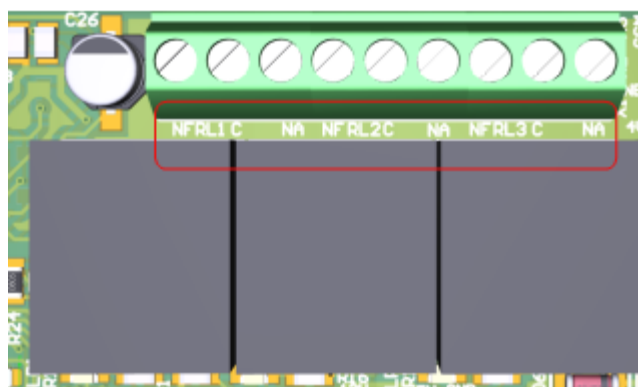
5.5 SAÍDAS A RELÉS

A CPB conta com 3 saídas a relés para acionamento de periféricos, tais como: motores, motobombas, lâmpadas, refletores, ventiladores etc.

5.5.1 Hardware



Pinagem dos Relés onboard.



Corrente e Tensão dos contatos:

20A – 14VDC

20A – 125VAC

10A – 220VAC

5.5.2 Software

Exemplo de Software para acionamento dos Relés da CPB.

```
rele_CPBinfinity | Arduino 1.8.10
File Edit Sketch Tools Help

#include <Wire.h>
#include "Adafruit_MCP23017.h"

Adafruit_MCP23017 mcp;

void setup()
{
  mcp.begin();

  mcp.pinMode(0, OUTPUT); // declaração de um pino de saída Relé 1
  mcp.pinMode(1, OUTPUT); // declaração de um pino de saída Relé 2
  mcp.pinMode(2, OUTPUT); // declaração de um pino de saída Relé 3
}

void loop()
{
  mcp.digitalWrite(0, HIGH); // liga a saída 0 do MCP (relé 1 da CPB)
  mcp.digitalWrite(0, LOW); // desliga a saída 0 do MCP (relé 1 da CPB)
  mcp.digitalWrite(1, HIGH); // liga a saída 1 do MCP (relé 2 da CPB)
  mcp.digitalWrite(1, LOW); // desliga a saída 1 do MCP (relé 2 da CPB)
  mcp.digitalWrite(2, HIGH); // liga a saída 2 do MCP (relé 3 da CPB)
  mcp.digitalWrite(2, LOW); // desliga a saída 2 do MCP (relé 3 da CPB)
}

Done compiling.
Sketch uses 2426 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 220 bytes (10%) of dynamic memory, leaving 1828 bytes for local variables.

18 Arduino/Genuine Uno on COM5
```

Exemplo “rele_CPBinfinity”, download disponível em:

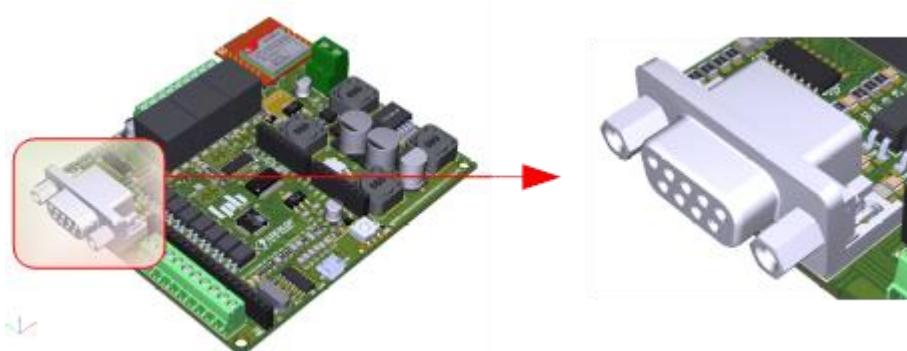
<https://github.com/casaautomacao/CPBinfinity>

5.6 COMUNICAÇÃO RS232 OU RS485

A CPB conta com uma saída para comunicação RS232 ou RS485, especializada para fazer a interface entre placa e IHM (Interface Homem-Máquina), CLP (Controlador Lógico Programável), Inversores de Frequência, etc.

Vídeo de Demonstração de funcionalidades da CPB com uma IHM Kinco, Inversor e Motor WEG: <https://www.youtube.com/watch?v=v4HOcDI71sM&t=4s>

5.6.1 Hardware



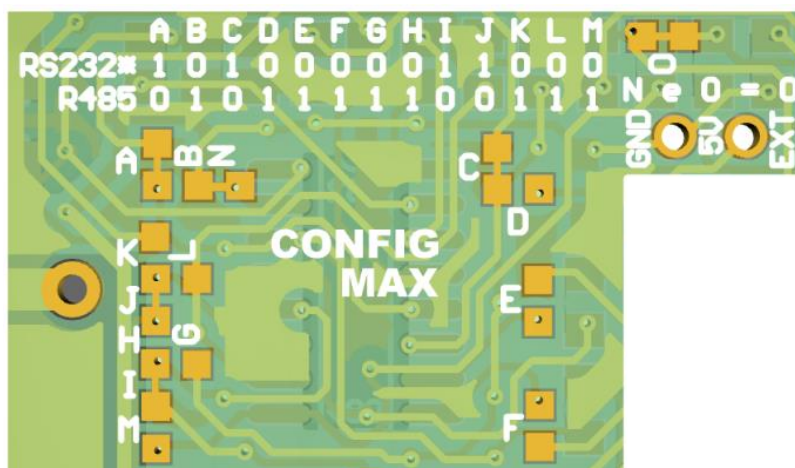
Pinagem de Conexão Serial:

Pino DB9	RS232*	RS485
1	-	B
2	TX	-
3	RX	-
4	-	-
5	GND	GND
6	-	A
7	-	-
8	-	-
9	-	-

Sendo RS232 o padrão do Hardware.

Configuração em Hardware da Comunicação Serial:

A tabela abaixo, mostra a matriz de configuração para definir se o protocolo de Comunicação Serial será RS232 ou RS485. Sendo RS232 default.

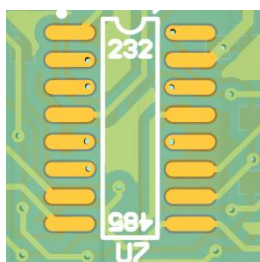


Onde 0 significa corte ou não solda e 1 significa solda nos jumpers A, B, C, D, E, F, G, H, I, J, K, L e M.

Os jumpers L e N são para alimentar externamente o Circuito para o MAX485, CI que fará a comunicação com protocolo RS485.

Comunicação Serial:	Função:	Configurar para:	Ação: — Corte — Solda
CONFIG MAX	RS232	RS485	

Cada protocolo de comunicação tem um CI específico, solde o mesmo seguindo as instruções serigrafadas na placa:



5.6.2 Software

Exemplo de Software para a comunicação RS232 da CPB com IHM's.



```
ihm_CPBinfinity | Arduino 1.8.10
File Edit Sketch Tools Help
ihm_CPBinfinity $ U_MODBUS
};

unsigned int holdingRegs[MB_REGS]; // function 3 and 16 register array

Tempora tempAcel;
Tempora Impressao;

void setup() {

  Serial.begin(9600);

  tempCom.defiSP(200);
  tempImpr.defiSP(1000);
  tempAcel.defiSP(20);
  Impressao.defiSP(500);

  modbus_configure(&Serial, 9600, SERIAL_8N2, 1, 13, MB_REGS, holdingRegs);

  modbus_update_comms(9600, SERIAL_8N2, 1);

}

void loop()
{

Invalid library found in C:\Users\Fábio\Documents\Arduino\libraries\AzureIoT: no headers
45 Arduino/Genuino Uno on COM5
```

Imagem somente do Setup devido a extensão do código, completo abaixo:

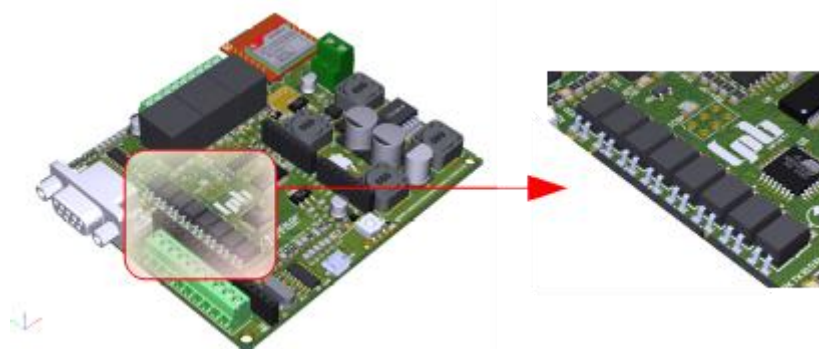
Exemplo “ihm_CPBinfinity”, download disponível em:

<https://github.com/casaautomacao/CPBinfinity>

5.7 OPTOACOPLOADORES

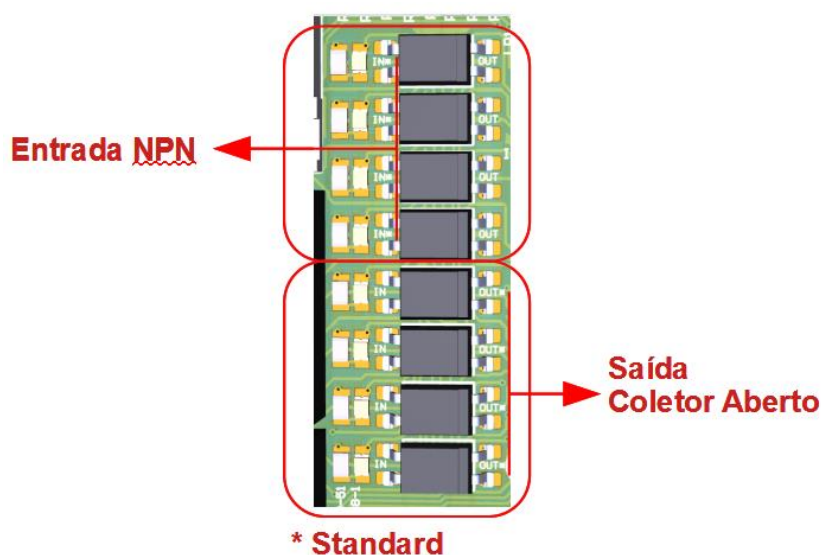
A CPB conta com 8 optoacopladores que podem ser configurados em hardware com Entradas NPN ou PNP ou saída Coletor Aberto, ideal para o acionamento ou leitura de periféricos que necessitem isolamento em relação ao Microcontrolador e/ou de fonte. Tendo assim, robustez nas leituras e acionamentos destas entradas e saídas.

5.7.1 Hardware



Default de Hardware são:

- 4 – Entradas NPN
- 4 – Saídas Coletor Aberto



Configuração em Hardware dos Optoacopladores:

A tabela abaixo, mostra a matriz de configuração para definir se os optoacopladores serão entradas PNP, NPN ou uma saída de Coletor Aberto.

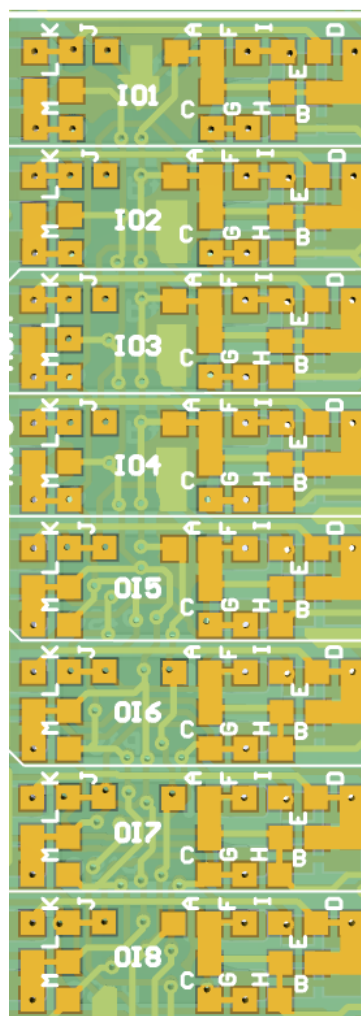
		A	B	C	D	E	F	G	H	I	J	K	L	M
INPUT PNP		0	0	1	0	1	0	0	1	1	0	1	0	1
INPUT NPN		1	0	0	0	1	0	1	0	1	0	1	0	1
OUTPUT		0	1	0	1	0	1	1	0	0	1	0	1	0
DEFAULT: 4 INPUT NPN 4 OUTPUT COLETOR ABERTO RESISTORES: 3K9 PARA INPUT 1K PARA OUTPUT														
CONFIG OPTOS														

Onde 0 significa corte ou não solda e 1 significa solda nos jumpers A, B, C, D, E, F, G, H, I, J, K, L e M da matriz abaixo.

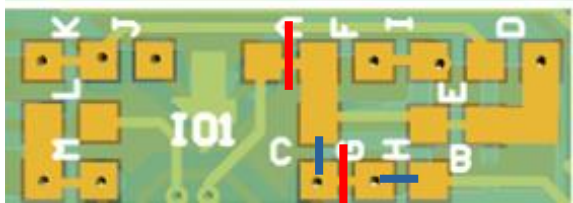
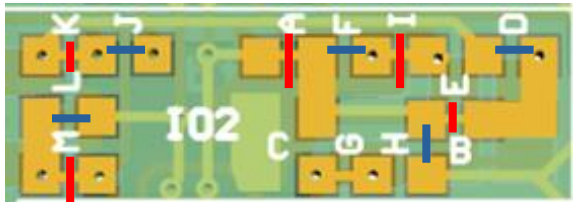
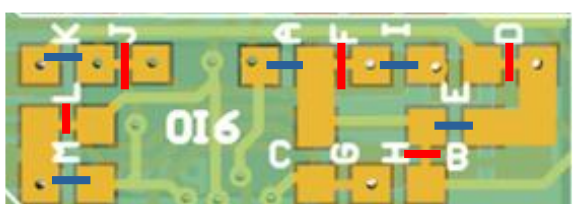
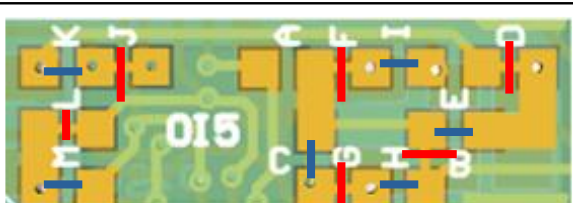
Default:

IO1-IO4 – Entrada NPN.

OI5-OI8 – Saída Coletor Aberto.



Para configurar os 8 optoacopladores da CPB siga as instruções da tabela abaixo:

Optoacopladores:	Função:	Configurar para:	Ação: — Corte — Solda
IO1, IO2, IO3 e IO4	Entrada NPN	Entrada PNP	
		Saída Coletor Aberto	
OI5, OI6, OI7 e OI8	Saída Coletor Aberto	Entrada NPN	
		Entrada PNP	

5.7.2 Software

Exemplo de Software para o acionamento ou leitura dos Optoacopladores.

```
optoacopladores_CPBinfinity | Arduino 1.8.10
File Edit Sketch Tools Help

optoacopladores_CPBinfinity $

#include <Wire.h>
#include "Adafruit_MCP23017.h"
Adafruit_MCP23017 mcp;

void setup() { // se a entrada for PNP, comentar as linhas do pullUp.
  mcp.begin();
  mcp.pinMode(5, INPUT);      // declaração de um pino de entrada NPN OPTOACOPLADA 1
  mcp.pullUp(5, HIGH);        // Aciona resistor de Pull Up interno.
  mcp.pinMode(6, INPUT);      // declaração de um pino de entrada NPN OPTOACOPLADA 2
  mcp.pullUp(6, HIGH);        // Aciona resistor de Pull Up interno.
  mcp.pinMode(7, INPUT);      // declaração de um pino de entrada NPN OPTOACOPLADA 3
  mcp.pullUp(7, HIGH);        // Aciona resistor de Pull Up interno.
  mcp.pinMode(8, INPUT);      // declaração de um pino de entrada NPN OPTOACOPLADA 4
  mcp.pullUp(8, HIGH);        // Aciona resistor de Pull Up interno.
  mcp.pinMode(9, OUTPUT);     // declaração de um pino de saída OPTOACOPLADA 5
  mcp.pinMode(10, OUTPUT);    // declaração de um pino de saída OPTOACOPLADA 6
  mcp.pinMode(11, OUTPUT);    // declaração de um pino de saída OPTOACOPLADA 7
  mcp.pinMode(12, OUTPUT);    // declaração de um pino de saída OPTOACOPLADA 8
}

void loop() {
  mcp.digitalRead(5); // leitura da entrada do multiplexador 5
  mcp.digitalRead(6); // leitura da entrada do multiplexador 6
  mcp.digitalRead(7); // leitura da entrada do multiplexador 7
  mcp.digitalRead(8); // leitura da entrada do multiplexador 7
  mcp.digitalWrite(9, HIGH); // exemplo de liga os pinos de saída do MCP9 ... 12
  mcp.digitalWrite(9, LOW);  // exemplo de desliga os pinos de saída do MCP9 ... 12
}

Auto Format finished.
Invalid library found in C:\Users\Fábio\Documents\Arduino\libraries\AzureIoT: no headers file.

4 Arduino/Genuino Uno on COM5
```

Exemplo “optoacopladores_CPBinfinity”, download disponível em:

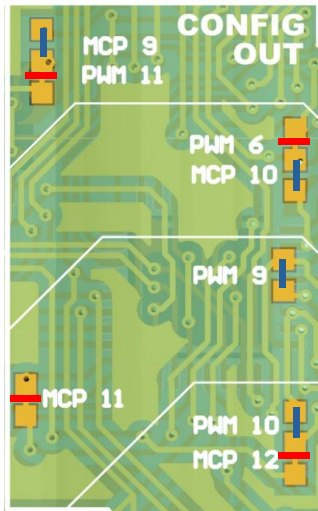
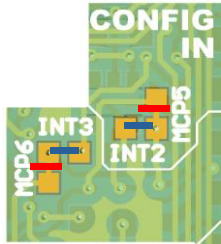
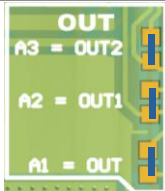
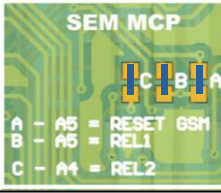
<https://github.com/casaautomacao/CPBinfinity>

6. CONFIGURAÇÕES INFINITY

A CPB Infinity conta com inúmeras opções de programação em Hardware, visando atender uma variedade maior de aplicações.

Na tabela abaixo estão as configurações adicionais de Hardware:

As configurações de Comunicação Serial e Optoacopladores estão nos itens 5.6 e 5.7

Grupo:	Função:	Configurar para:	Ação: — Corte — Solda
CONFIG OUT	Saída Optoacoplada Multiplexada 9	Saída Optoacoplada PWM 11 do uC	
	Saída Optoacoplada Multiplexada 10	Saída Optoacoplada PWM 6 do uC	
	Saída Optoacoplada Multiplexada 11	Saída Optoacoplada PWM 9 do uC	
	Saída Optoacoplada Multiplexada 12	Saída Optoacoplada PWM 10 do uC	
CONFIG IN	Entrada Optoacoplada Multiplexada 5	Entrada Optoacoplada Interrupção 2 do uC	
	Entrada Optoacoplada Multiplexada 6	Entrada Optoacoplada Interrupção 3 do uC	
OUT	Pino Analógico A1	Barramento OUT	
	Pino Analógico A2	Barramento OUT1	
	Pino Analógico A3	Barramento OUT2	
SEM MCP	C - Pino Analógico A4	Acionar Relé 2	
	B - Pino Analógico A5	Acionar Relé 1	
	A - Pino Analógico A5	Reset GSM	
AD	T - Pino Analógico A0	Acionar Relé 3	
AND	Sem Comunicação Serial	Com Comunicação Serial	



7. CONCLUSÃO

Nos conheça e que juntos, possamos desenvolver produtos tecnológicos, inovadores, desafiadores e rentáveis.



VAMOS CONVERSAR?

Entre em contato conosco para conferir muito mais sobre nossa empresa. Será um prazer atendê-lo!

 comercial@crescerengenharia.com

 51 3239-0553

 crescerengenharia.com

 Crescer Indústria de Automação

 A Casa da Automação

 @crescer_automacao

 /casadaautomacao