

# Crescer Professional Board Infinity

Manual de Utilização

## SUMÁRIO

<b>1. INFORMAÇÕES GERAIS .....</b>	<b>4</b>
<b>2. SOBRE A CPB .....</b>	<b>4</b>
<b>3. HARDWARE.....</b>	<b>5</b>
<b>3.1 PINAGEM.....</b>	<b>6</b>
<b>3.2 APLICAÇÕES .....</b>	<b>7</b>
<b>4. SOFTWARE .....</b>	<b>8</b>
<b>5. ESPECIFICAÇÕES DAS FUNCIONALIDADES .....</b>	<b>9</b>
<b>5.1 FONTE DE ALIMENTAÇÃO .....</b>	<b>9</b>
5.1.1 Hardware.....	9
<b>5.2 COMUNICAÇÃO GSM .....</b>	<b>10</b>
5.2.1 Hardware.....	10
5.2.2 Software .....	11
<b>5.3 MULTIPLEXADOR .....</b>	<b>13</b>
5.3.1 Hardware.....	13
5.3.2 Software .....	14
<b>5.4 BARRAMENTO .....</b>	<b>15</b>
5.4.1 Hardware.....	15
5.4.2 Software .....	16
<b>5.5 SAÍDAS A RELÉS.....</b>	<b>17</b>
5.5.1 Hardware.....	17
<b>5.6 COMUNICAÇÃO RS232 ou RS485 .....</b>	<b>19</b>
5.6.1 Hardware.....	19
5.6.2 Software .....	21
<b>5.7 OPTOACOPLADORES.....</b>	<b>22</b>
5.7.1 Hardware.....	22
5.7.2 Software .....	25

<b>6. CONFIGURAÇÕES INFINITY .....</b>	<b>26</b>
<b>7. CONCLUSÃO .....</b>	<b>27</b>



## 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=KqLikDG5SNE>

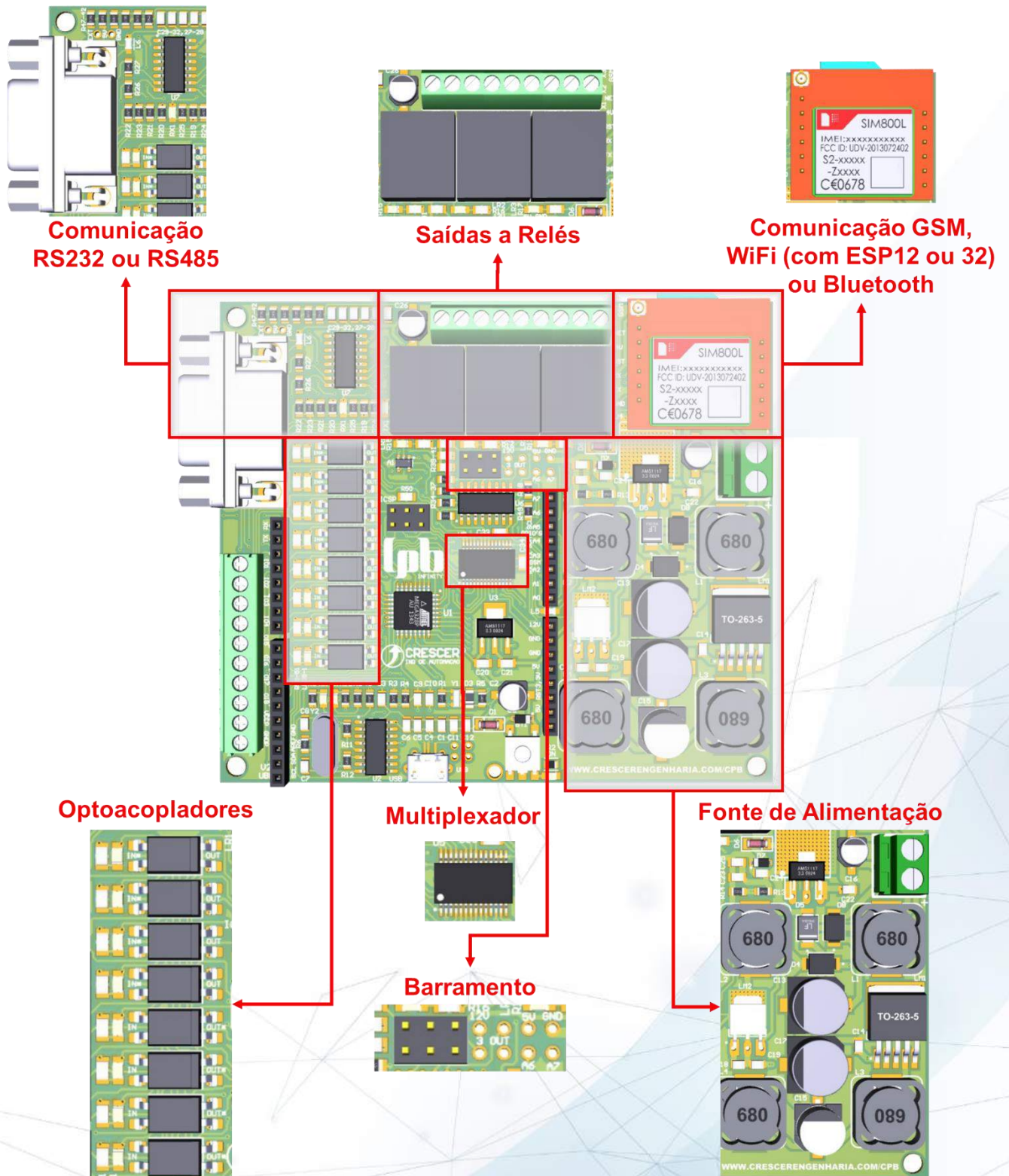
A CPB utiliza o Microcontrolador Atmega328p e Interface de Desenvolvimento (IDE) do Arduino para programação.

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

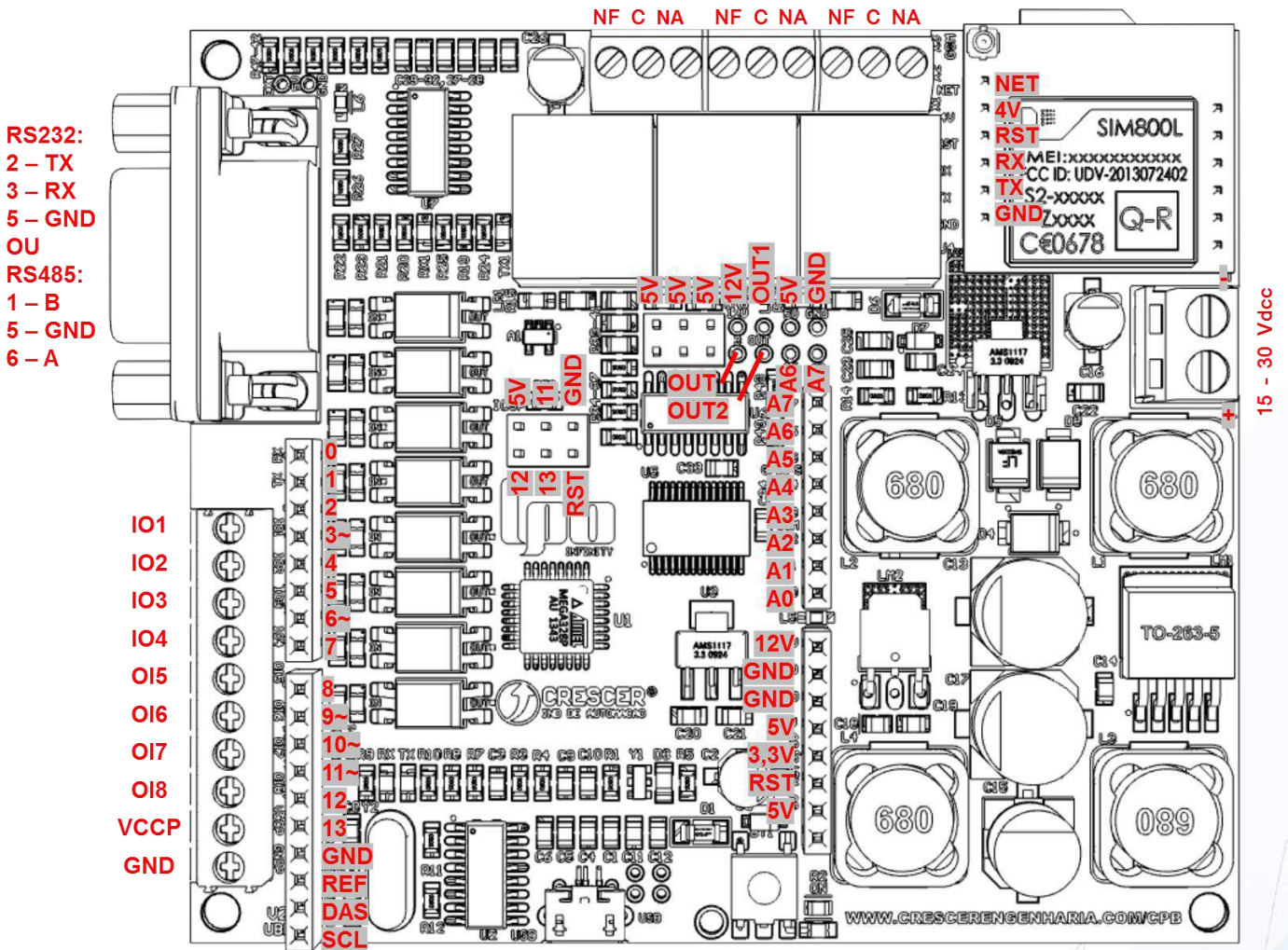


### 3. HARDWARE

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	Relés
RST	Reset do Microcontrolador	
NF	Normalmente Fechado	
NA	Normalmente Aberto	
C	Comum	Externos
RL1, RL2, RL3	Relés	
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.

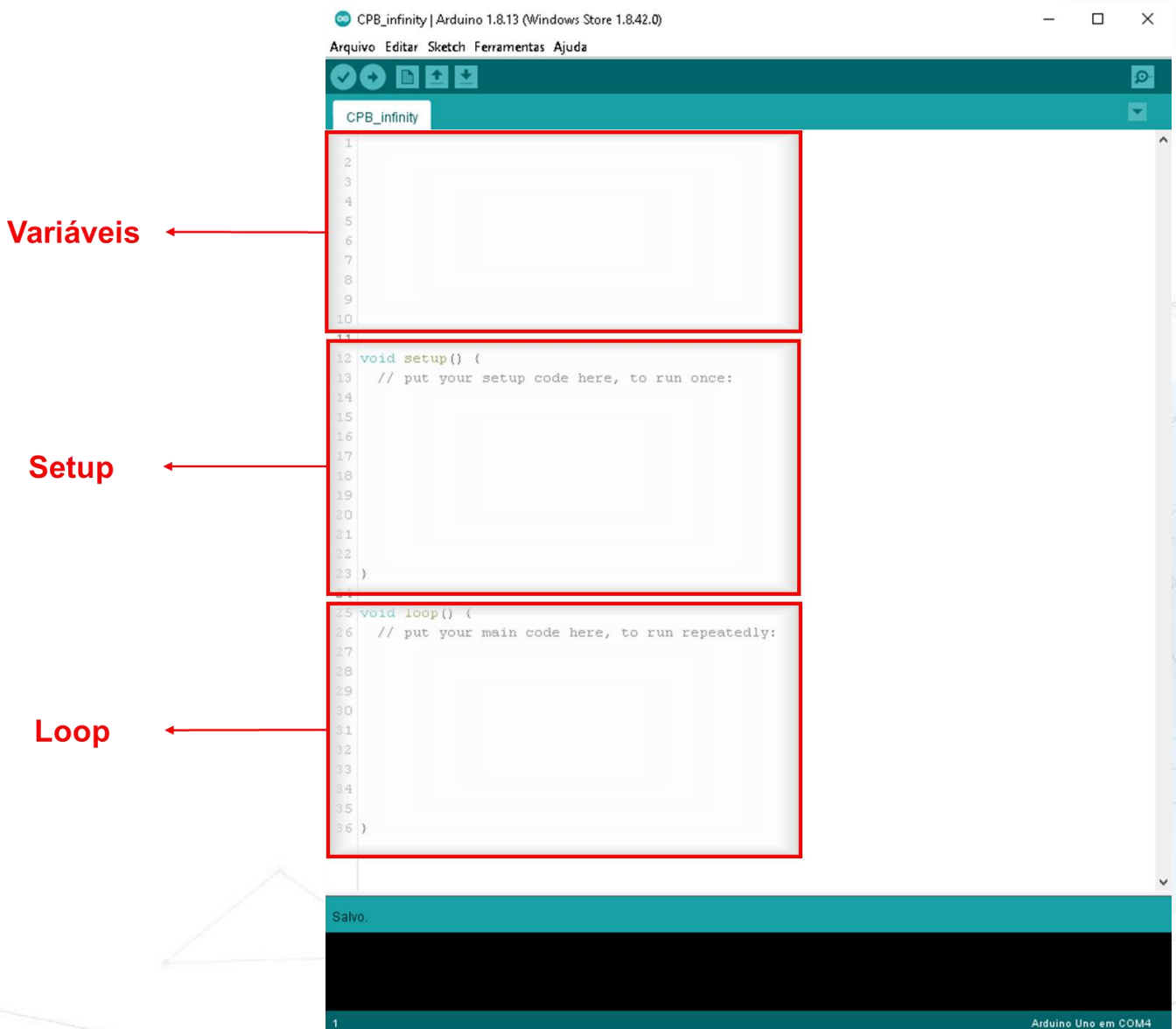


## 4. SOFTWARE

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>





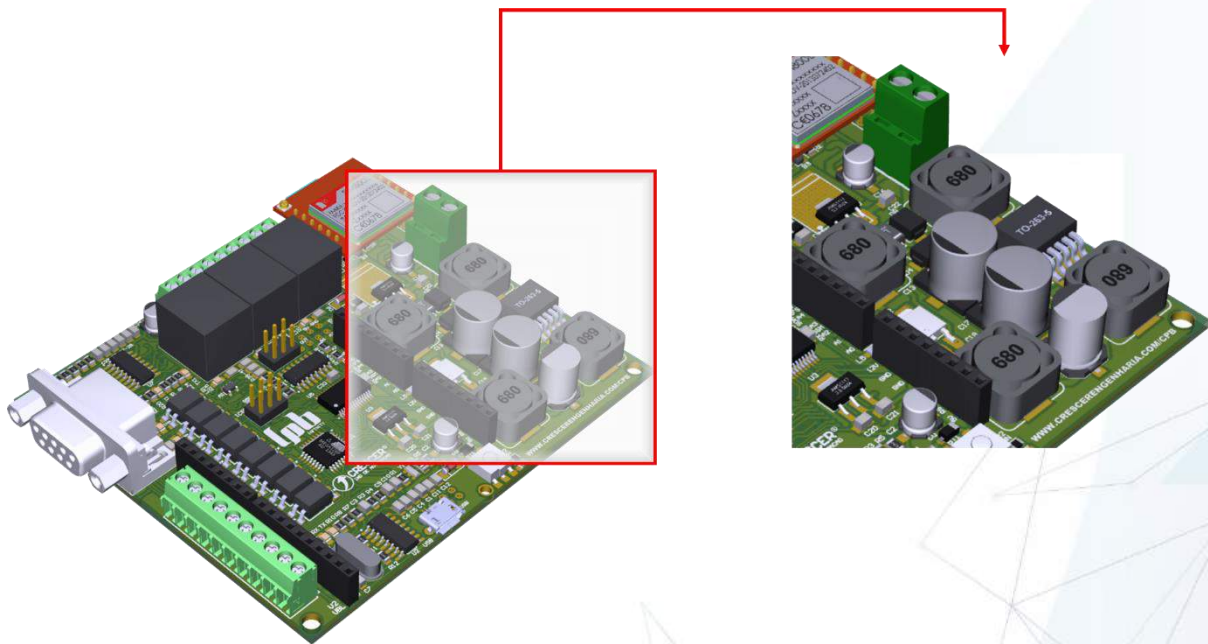
## 5. ESPECIFICAÇÕES DAS FUNCIONALIDADES

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.

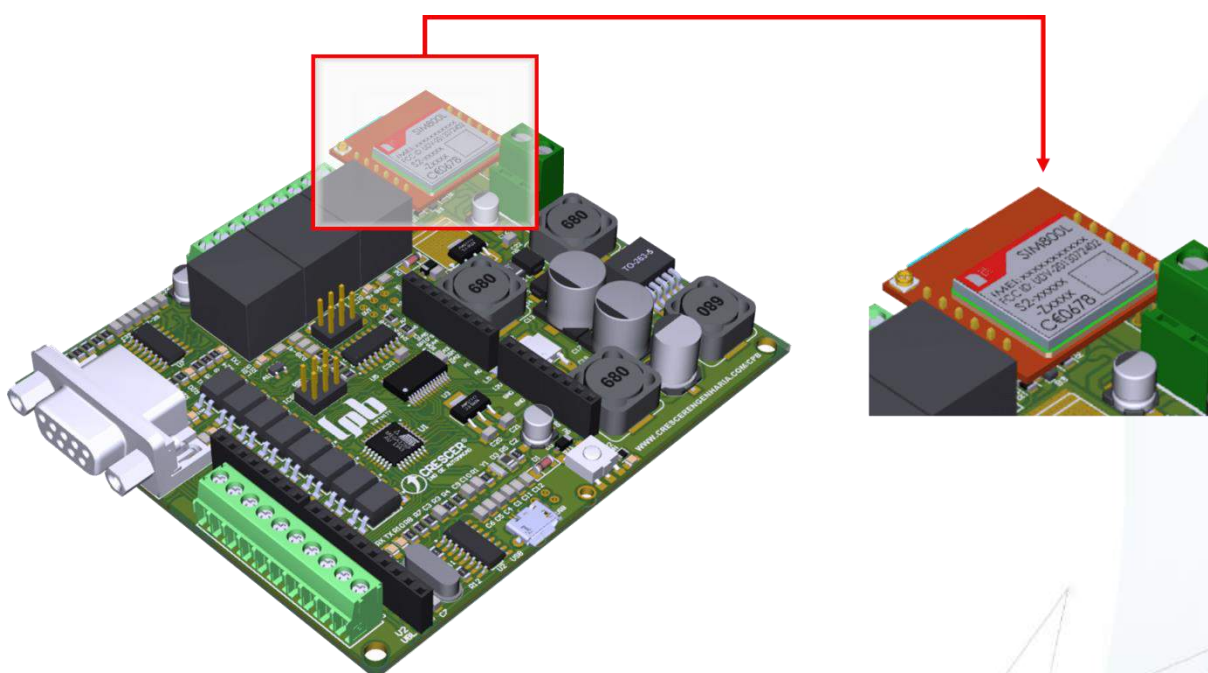
Consumo: 6 W.

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:



The screenshot shows the Arduino IDE interface with the following code visible in the editor:

```
1 #include <SoftwareSerial.h>
2 SoftwareSerial gsm(A3, A2); // RX, TX
3 #include <Sim8001.h>
4 Sim8001 Sim8001;
5
6 #include <Crescer.h>
7
8 Tempora temp1;
9
10 String avalia = "";
11 int conta = 0;
12
13 void setup()
14 {
15   Serial.begin(9600);
16   delay(1000);
17
18   gsm.begin(9600);
19   gsm.listen();
20
21   comando ("AT");
22   comando ("AT+CMGF=1");
23   comando ("ATE0");
24   delay(1500);
25   comando ("AT+CNMI=1,2,0,0,0");
26   delay(1000);
27   temp1.defiSP(500);
28   delay(8000);
29 }
30
31 int teste;
32
33 void loop()
34 {
35   String comp = "";
36   if (gsm.available() > 0) {
37
38     String msggsm = "";
39     while (gsm.available() > 0)
40     {
41       char letra = gsm.read();
42       if (isAlphaNumeric(letra)) msggsm += letra;
43     }
44     int tam = msggsm.length();
```

The IDE window title is "gsm\_CPBinfinity | Arduino 1.8.13 (Windows Store 1.8.42.0)". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". The toolbar shows icons for saving, running, and other IDE functions. The status bar at the bottom indicates "34" and "Arduino Uno em COM4".

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>

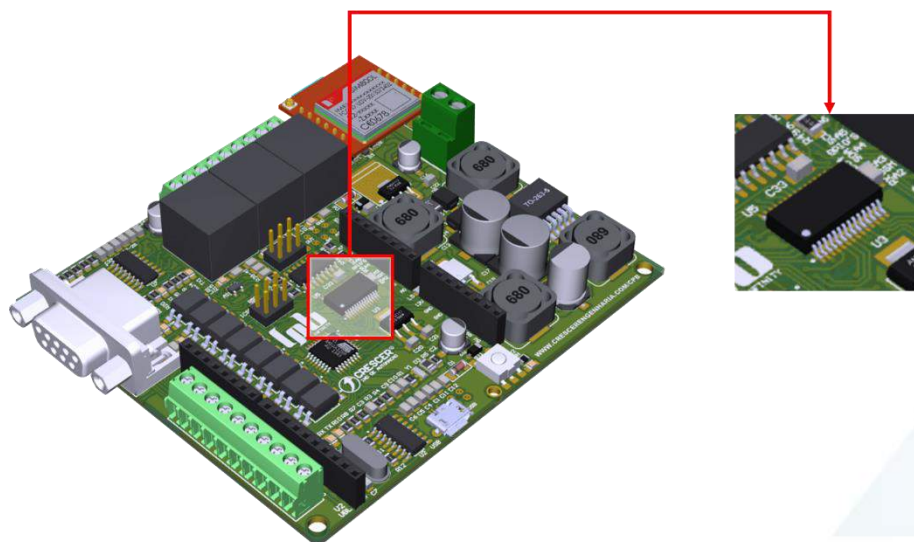
Parte 2 - [https://www.youtube.com/watch?v=yXZ\\_SBmgF8w](https://www.youtube.com/watch?v=yXZ_SBmgF8w)

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

## 5.3 MULTIPLEXADOR

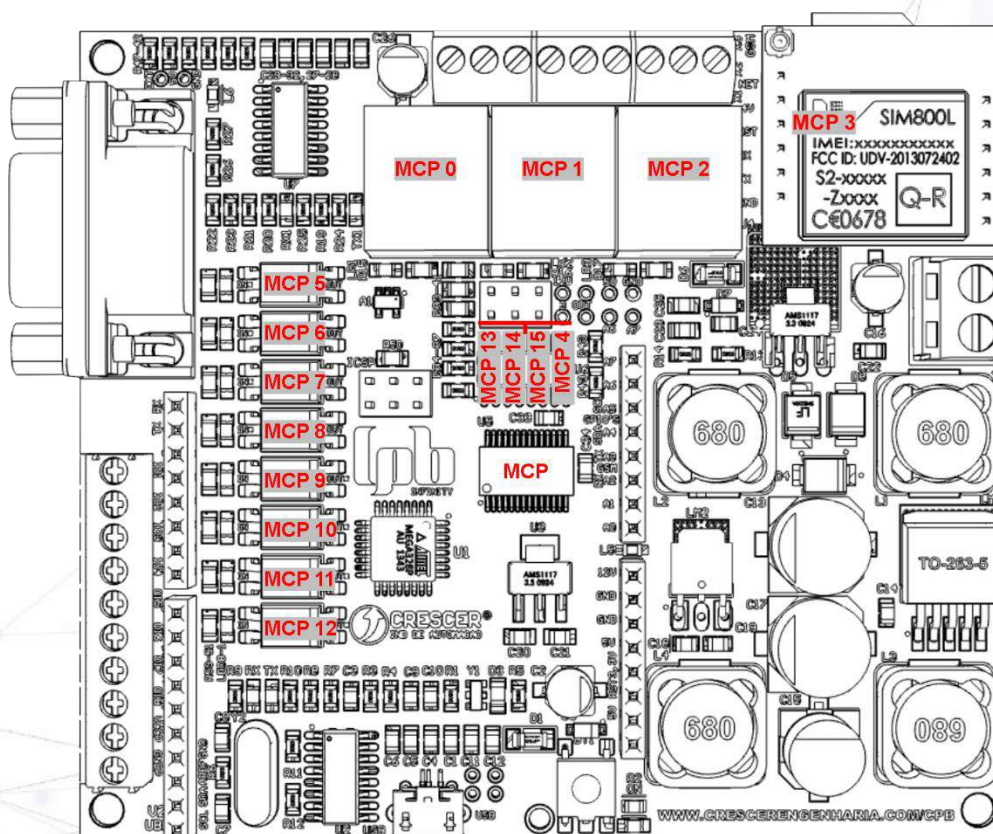
A CPB conta com um Chip Multiplexador para utilizar menos pinos do Microcontrolador e ter um aumento na quantidade disponíveis de I/O's.

### 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 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

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 Relé 1
12 mcp.pinMode(1, OUTPUT); // declaração de um pino de saída Relé 2
13 mcp.pinMode(2, OUTPUT); // declaração de um pino de saída Relé 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
29 mcp.pinMode(14, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO
30 mcp.pinMode(15, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO
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 }

```

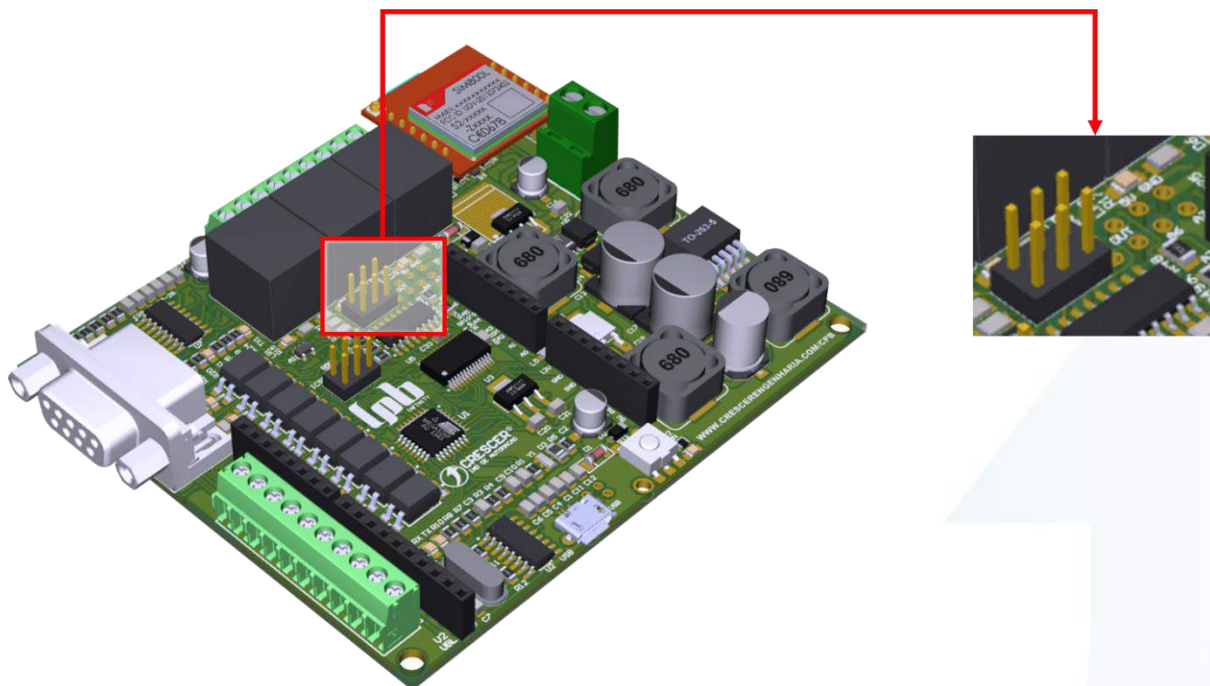
Exemplo “multiplexador\_CPBinfinity”, download disponível em: <https://github.com/casaautomacao/CPBinfinity>



## 5.4 BARRAMENTO

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

### 5.4.1 Hardware

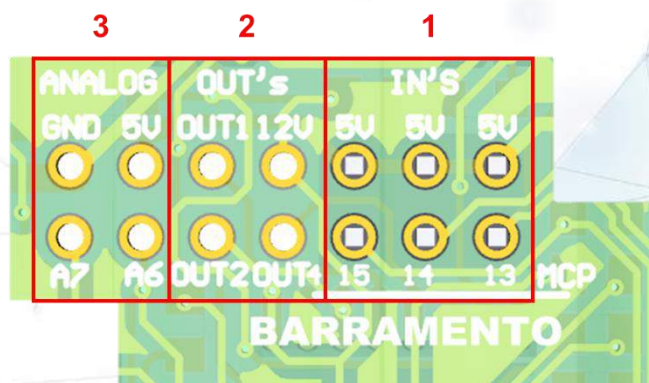


O barramento possui:

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:

```

multiplexador_CPBinfinity | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

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 Relé 1
12 mcp.pinMode(1, OUTPUT); // declaração de um pino de saída Relé 2
13 mcp.pinMode(2, OUTPUT); // declaração de um pino de saída Relé 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
29 mcp.pinMode(14, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO
30 mcp.pinMode(15, INPUT); // declaração de um pino de entrada contato seco no BARRAMENTO
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 }

```

1 Arduino Uno em COM4

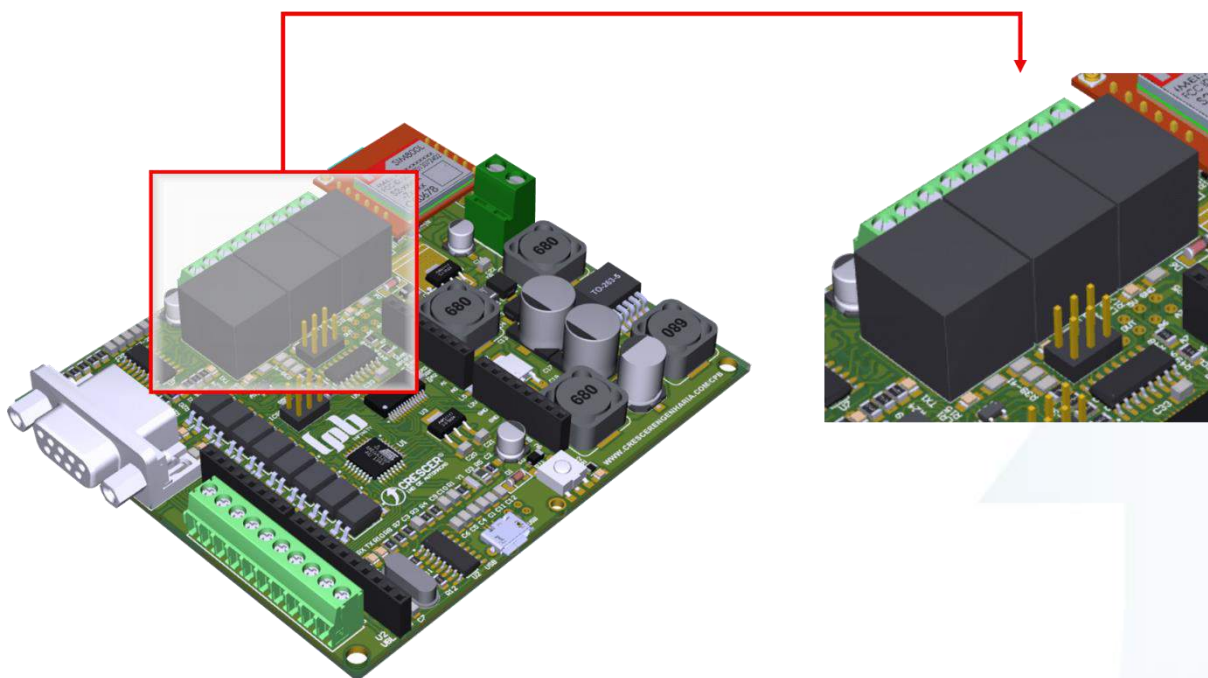
Exemplo “barramento\_CPBinfinity”, download disponível em:

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

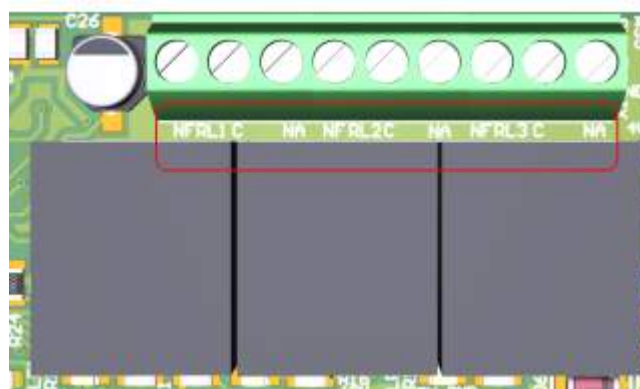
## 5.5 SAÍDAS A RELÉS

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 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

rele_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
12 mcp.pinMode(0, OUTPUT); // declaração de um pino de saída Relé 1
13 mcp.pinMode(1, OUTPUT); // declaração de um pino de saída Relé 2
14 mcp.pinMode(2, OUTPUT); // declaração de um pino de saída Relé 3
15
16 }
17
18 void loop()
19 {
20
21 mcp.digitalWrite(0, HIGH); // liga a saída 0 do MCP (relé 1 da CPB)
22 mcp.digitalWrite(0, LOW); // desliga a saída 0 do MCP (relé 1 da CPB)
23 mcp.digitalWrite(1, HIGH); // liga a saída 1 do MCP (relé 2 da CPB)
24 mcp.digitalWrite(1, LOW); // desliga a saída 1 do MCP (relé 2 da CPB)
25 mcp.digitalWrite(2, HIGH); // liga a saída 2 do MCP (relé 3 da CPB)
26 mcp.digitalWrite(2, LOW); // desliga a saída 2 do MCP (relé 3 da CPB)
27
28 }

1 Arduino Uno em COM4
```

Exemplo “rele\_CPBinfinity”, download disponível em:

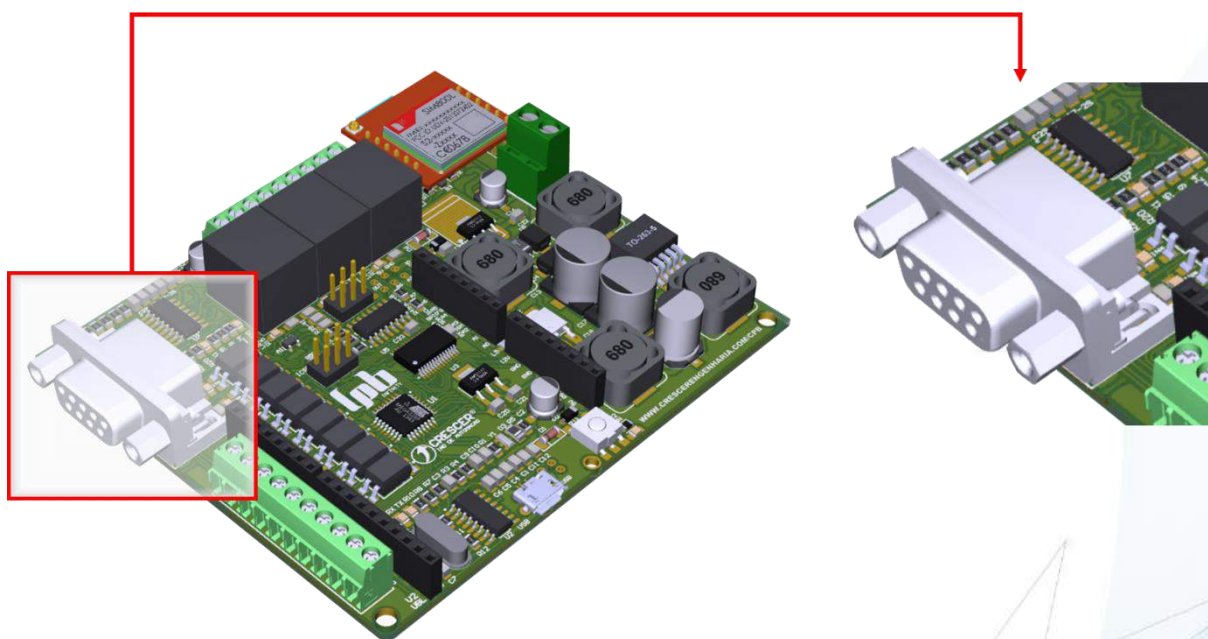
<https://github.com/casaautomacao/CPBinfinity>

## 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



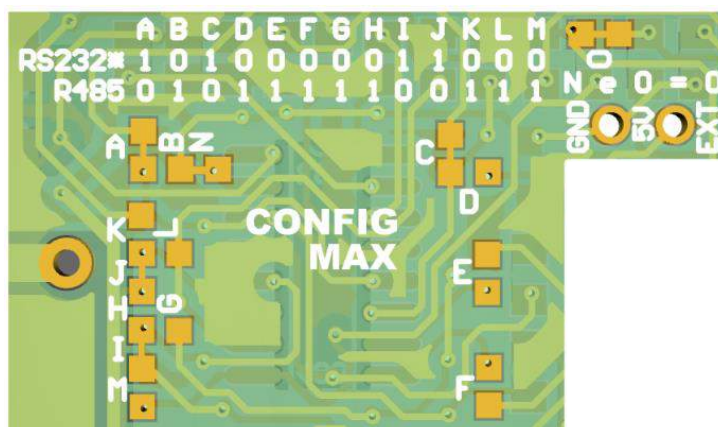
Pinagem de Conexão Serial:

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.

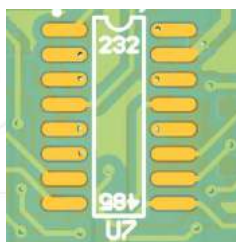


Onde 0 significa corte ou não solda e 1 significa solda nos jumpers A, B, C, D, E, F, G, H, I, J, K, L e M.

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: <span style="color: red;">—</span> Corte <span style="color: blue;">—</span> Solda
CONFIG MAX	RS232	RS485	

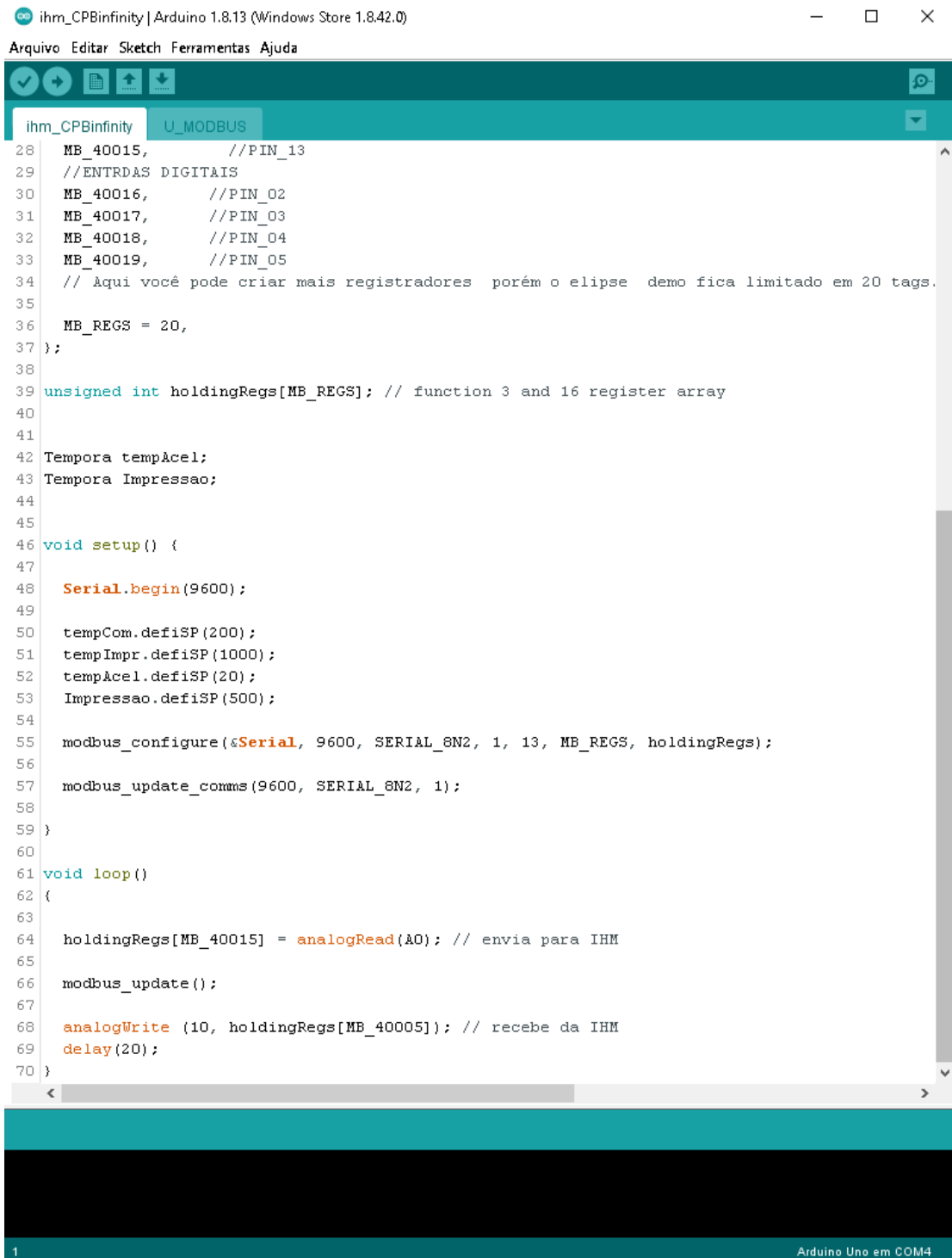
Cada protocolo de comunicação tem um CI específico, solde o mesmo seguindo as instruções serigrafadas na placa:





## 5.6.2 Software

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



```

ihm_CPBinfinity | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

ihm_CPBinfinity U_MODBUS
28 MB_40015, //PIN_13
29 //ENTR&AS DIGITAIS
30 MB_40016, //PIN_02
31 MB_40017, //PIN_03
32 MB_40018, //PIN_04
33 MB_40019, //PIN_05
34 // Aqui voc&eacute; pode criar mais registradores por&eacute;m o elipse demo fica limitado em 20 tags.
35
36 MB_REGS = 20,
37 };
38
39 unsigned int holdingRegs[MB_REGS]; // function 3 and 16 register array
40
41
42 Tempora temp&acel;
43 Tempora Impressao;
44
45
46 void setup() {
47
48     Serial.begin(9600);
49
50     tempCom.defiSP(200);
51     tempImpr.defiSP(1000);
52     temp&acel.defiSP(20);
53     Impressao.defiSP(500);
54
55     modbus_configure(&Serial, 9600, SERIAL_8N2, 1, 13, MB_REGS, holdingRegs);
56
57     modbus_update_comms(9600, SERIAL_8N2, 1);
58
59 }
60
61 void loop()
62 {
63
64     holdingRegs[MB_40015] = analogRead(A0); // envia para IHM
65
66     modbus_update();
67
68     analogWrite (10, holdingRegs[MB_40005]); // recebe da IHM
69     delay(20);
70 }
1 Arduino Uno em COM4

```

Imagem somente do Setup devido a extens&oatilde;o do c&odig;mo, completo abaixo:

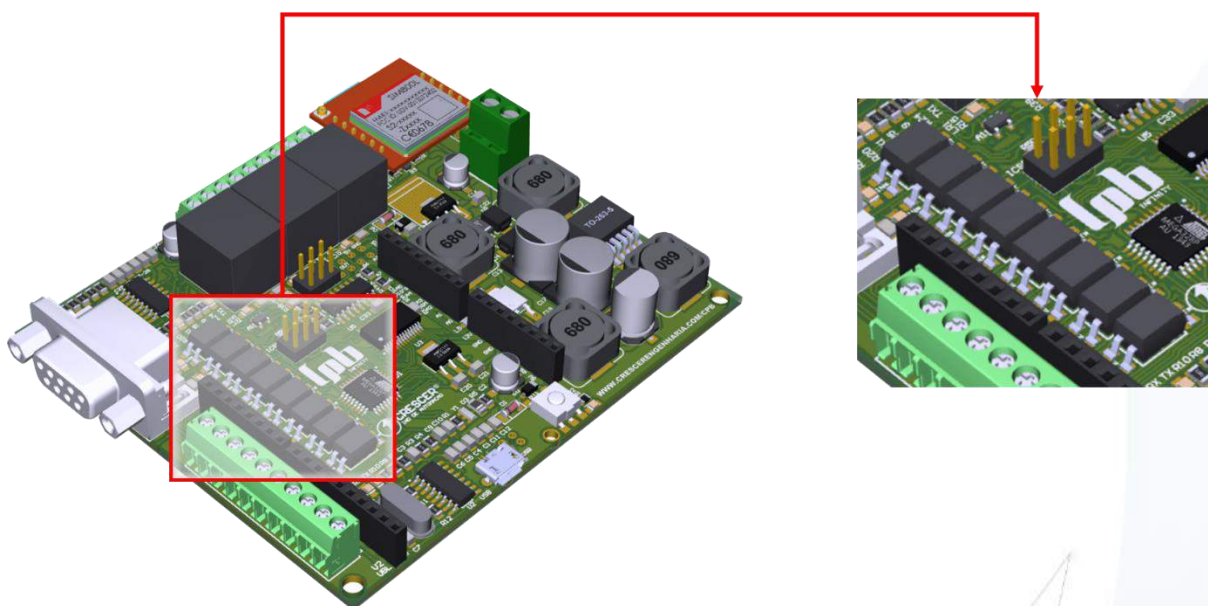
Exemplo "ihm\_CPBinfinity", download dispon&iv&eacute;l em:

<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.

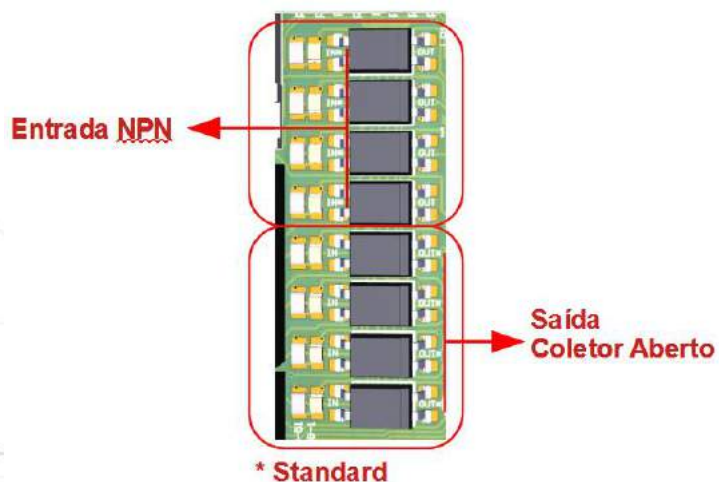
### 5.7.1 Hardware



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.

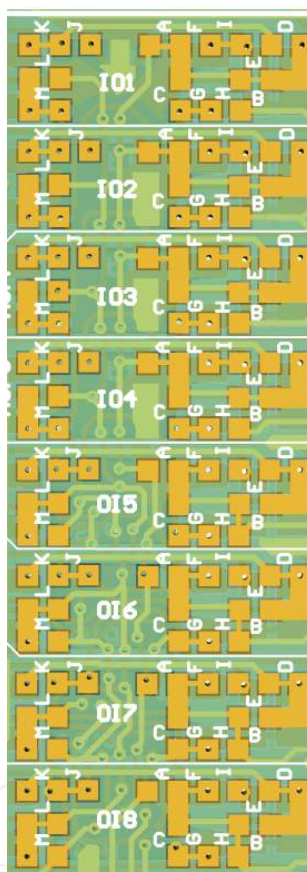
	A	B	C	D	E	F	G	H	I	J	K	L	M
INPUT PNP	0	0	1	0	1	0	0	1	1	0	1	0	1
INPUT NPN	1	0	0	0	1	0	1	0	1	0	1	0	1
OUTPUT	0	1	0	1	0	1	1	0	0	1	0	1	0

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.*

*IO5-IO18 – Saída Coletor Aberto.*



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

Optoacopladores:	Função:	Configurar para:	Ação: <span style="color: red;">— Corte</span> <span style="color: blue;">— Solda</span>
IO1, IO2, IO3 e IO4	Entrada NPN	Entrada PNP	
		Saída Coletor Aberto	
O15, O16, O17 e O18	Saída Coletor Aberto	Entrada NPN	
		Entrada PNP	

## 5.7.2 Software

Exemplo de Software para o acionamento ou leitura dos Optoacopladores.



```

optoacopladores_CPBinfinity | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

optoacopladores_CPBinfinity
1 #include <Wire.h>
2
3 #include "Adafruit_MCP23017.h"
4
5 Adafruit_MCP23017 mcp;
6
7
8 void setup() { // se a entrada for PNP, comentar as linhas do pullUp.
9   mcp.begin();
10  mcp.pinMode(5, INPUT); // declaração de um pino de entrada NPN OPTOACOPLADA 1
11  mcp.pullUp(5, HIGH); // Aciona resistor de Pull Up interno.
12  mcp.pinMode(6, INPUT); // declaração de um pino de entrada NPN OPTOACOPLADA 2
13  mcp.pullUp(6, HIGH); // Aciona resistor de Pull Up interno.
14  mcp.pinMode(7, INPUT); // declaração de um pino de entrada NPN OPTOACOPLADA 3
15  mcp.pullUp(7, HIGH); // Aciona resistor de Pull Up interno.
16  mcp.pinMode(8, INPUT); // declaração de um pino de entrada NPN OPTOACOPLADA 4
17  mcp.pullUp(8, HIGH); // Aciona resistor de Pull Up interno.
18  mcp.pinMode(9, OUTPUT); // declaração de um pino de saída OPTOACOPLADA 5
19  mcp.pinMode(10, OUTPUT); // declaração de um pino de saída OPTOACOPLADA 6
20  mcp.pinMode(11, OUTPUT); // declaração de um pino de saída OPTOACOPLADA 7
21  mcp.pinMode(12, OUTPUT); // declaração de um pino de saída OPTOACOPLADA 8
22 }
23 void loop() {
24  mcp.digitalRead(5); // leitura da entrada do multiplexador 5
25  mcp.digitalRead(6); // leitura da entrada do multiplexador 6
26  mcp.digitalRead(7); // leitura da entrada do multiplexador 7
27  mcp.digitalRead(8); // leitura da entrada do multiplexador 7
28  mcp.digitalWrite(9, HIGH); // exemplo de liga os pinos de saída do MCP9 ... 12
29  mcp.digitalWrite(9, LOW); // exemplo de desliga os pinos de saída do MCP9 ... 12
30
31 }
  
```

Salvo.

1 Arduino Uno em COM4

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: <span style="color: red;">—</span> Corte <span style="color: blue;">—</span> 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.

Indicações de materiais adicionais:

- 1 – Versão CPB Infinity e CPB 32: [Quero conhecer!](#)
- 2 – Blog Crescer com conteúdo de Arduino Profissional, ESP32, Internet Of Things, Eletrotécnica e Engenharia: [Quero saber mais!](#)
- 3 – Grupo no WhatsApp “Profissionais com Arduino”: [Quero participar!](#)
- 4 – Módulo I gratuito do curso “Automação Profissional com Arduino”: [Quero aprender!](#)
- 5 - Drive do banco de códigos para serem utilizados nas CPB`s: [Quero utilizar!](#)

## ANEXO I – DETALHES TÉCNICOS CPB'S

			
INFORMAÇÕES	CPB INFINITY	CPB WIFI 32	CPB MEGA
Relés 10A NA e NF		3	7
Optoacopladores de Entrada (NPN OU PNP) *1		4	9
Optoacopladores de Saídas (Coletor Aberto) *1		4	9
IO Digitais na Barra de Pinos *2		14	
Saída PWM	6	12	12
Entradas Analógicas	8	12	16
Saídas Analógicas	0	2	0
Resolução Analógica	10 Bits	12 Bits	10 Bits
Integração com Shields Arduino		SIM	
Comunicação I2C		SIM	
Comunicação SPI		SIM	
Comunicação Serial UART	1	3	4
Comunicação RS232 ou RS485		SIM	
BLUETOOTH	Com SIM800C ou Esp32	On Board	Com SIM800C ou Esp32
BLE	Com Esp32		Com Esp32
WIFI	Com ESP12 ou Esp32		Com ESP12 ou Esp32
GPRS		Com Shield SIM800C ou SIM800L	
GSM			
Espaço para Prototipagem com 25 IOs disponíveis		NÃO	SIM
Alimentação 15-30 VCC	770 mW	380 mW	1300 mW
* 1: Os Optoacopladores podem ser configurados tanto como Saída coletor aberto ou como Entrada PNP ou NPN.			
* 2: Grande parte das IOs são multifuncionais, estando acima o máximo disponível e a composição entre elas dependerá da aplicação.			
* 3: Shields como:			
Internet cabeada: 	USB Host: 	Sd Card: 	IHM Nxtion: 