

**Placa de expansão multifunções compatível com
Arduino (ID PR.FT11_5)**

Guia

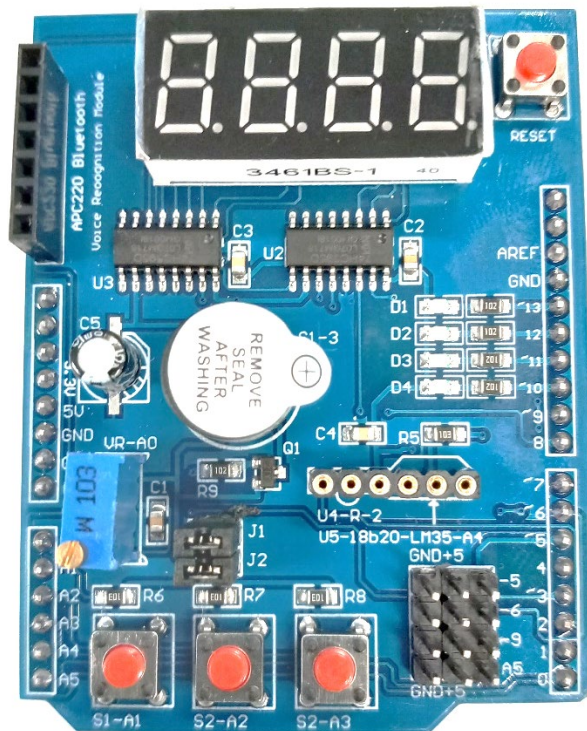
V 1.0

Índice

1. Introdução.....	3
2. Características.....	4
3. Especificações técnicas	5
4. Exemplos de código	7
5. Diagrama do circuito.....	16

1. Introdução

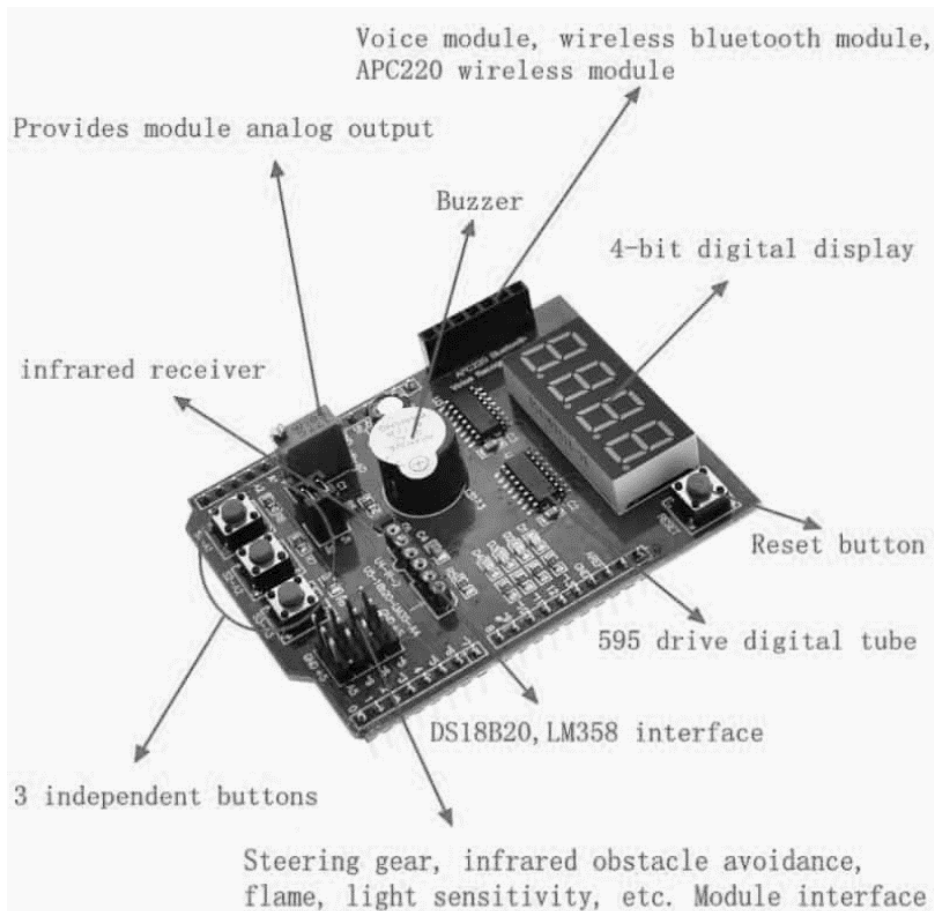
O manual refere-se a uma de placa de expansão multifunções compatível com Arduino. Esta placa está preparada para desempenhar diversas funções.



2. Características

- 1 Módulo visor LED de 7 segmentos e 4 dígitos
- 4 LEDs montados à superfície em configuração paralela
- 1 Potenciômetro de precisão, regulável até 10K
- 3 botões de pressão independentes
- 1 Campainha “piezoelétrica”
- 1 interface disponível DS18B20 e LM35
- 1 interface recetor por infravermelhos
- 1 interface de ligação série

3. Especificações técnicas



Instalação das bibliotecas da placa multifuncional

TimerOne

<https://github.com/PaulStoffregen/TimerOne>

Software I2C

<https://playground.arduino.cc/Main/SoftwareI2CLibrary/>

ou

<https://github.com/felias-fogg/SoftI2CMaster>

Se, por algum motivo, não conseguir instalar nenhuma das bibliotecas, faça o download, descompacte e copie para a pasta Documents\Arduino\libraries (se estiver a usar o Windows).

4. Exemplos de código

Exemplo 1 – Como usar o buzzer da placa

```
#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>

void setup() {
  Serial.begin(9600);
  MFS.initialize();
  MFS.beep();
  delay(1000);
  MFS.beep(5,5,4,3,50);
}

void loop() {
}
```

Exemplo 2 – Detecção de botões pressionados

Diferentes tipos de pressionamento de botão podem ser detetados. Curto, longo, liberação do botão após pressionamento curto, liberação do botão após pressionamento longo.

```
#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>

void setup() {
  Serial.begin(9600);
  MFS.initialize();
}

void loop() {
  byte btn = MFS.getButton();
  if (btn)
```

```

{
  byte buttonNumber = btn & B00111111;
  byte buttonAction = btn & B11000000;

  Serial.print("BUTTON_");
  Serial.write(buttonNumber + '0');
  Serial.print("_");
  if (buttonAction == BUTTON_PRESSED_IND)
  {
    Serial.println("PRESSED");
  }
  else if (buttonAction == BUTTON_SHORT_RELEASE_IND)
  {
    Serial.println("SHORT_RELEASE");
  }
  else if (buttonAction == BUTTON_LONG_PRESSED_IND)
  {
    Serial.println("LONG_PRESSED");
  }
  else if (buttonAction == BUTTON_LONG_RELEASE_IND)
  {
    Serial.println("LONG_RELEASE");
  }
}
}
}

```

Exemplo 3 - Visor de dígitos

```

#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>

void setup() {
  Serial.begin(9600);
  Timer1.initialize();
  MFS.initialize(&Timer1);
  MFS.write("Hi");
  delay(2000);
  MFS.write(-273);
}

```



```

delay(2000);
MFS.write(3.141, 2);
delay(2000);
}
int counter=0;
byte ended = false;

void loop() {
if (counter < 200)
{
MFS.write((int)counter); counter++;
}
else if (!ended)
{
ended = true;
MFS.write("End");
MFS.blinkDisplay(DIGIT_ALL, ON);
}
delay(50);
}

```

Exemplo 4 – Controlo de LEDs

```

#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>

void setup() {
Serial.begin(9600);
Timer1.initialize();
MFS.initialize(&Timer1);

MFS.writeLeds(LED_ALL, ON);
delay(2000);
MFS.blinkLeds(LED_1 | LED_2, ON);
delay(2000);
MFS.blinkLeds(LED_1 | LED_2, OFF);
}

```

```
MFS.blinkLeds(LED_3 | LED_4, ON);
delay(2000);
MFS.blinkLeds(LED_ALL, ON);
delay(2000);
MFS.blinkLeds(LED_ALL, OFF);
MFS.writeLeds(LED_ALL, OFF);
}

void loop() {
}
```

Exemplo 5 – Ler o valor do potenciômetro

```
#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>

void setup() {
  Serial.begin(9600);
  MFS.initialize(&Timer1);
}

void loop() {
  MFS.write(analogRead(POT_PIN));
  delay(100);
}
```

Exemplo 5 – Contagem de pulsos

A placa multifuncional tem suporte para contagem de pulsos (até 500Hz) aplicados a um pino de entrada do Arduino. A contagem de pulsos é gerida em segundo plano por meio de interrupções, o que permite que o programa se

concentre na execução de sua tarefa principal. Pressione repetidamente o botão 1 para gerar pulsos e veja a leitura da taxa de pressão no visor de dígitos.

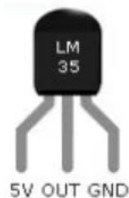
```
#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>

void setup() {
  Serial.begin(9600);
  MFS.initialize();
  MFS.initPulseInCounter( BUTTON_1_PIN,1500, LOW);
}

void loop() {
  unsigned int pulsePeriodMs = MFS.getPulseInPeriod();
  if (pulsePeriodMs == 0)
  {
    MFS.write(0.0, 1);
  }
  else
  {
    MFS.write(1000.0 / pulsePeriodMs, 1);
  }
}
```

Exemplo 6 – Ler temperatura do sensor LM35

A placa tem suporte para o sensor de temperatura LM35, que deve ser inserido corretamente, caso contrário, poderá danificar o Arduino e o sensor ficará muito quente!



```

#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>

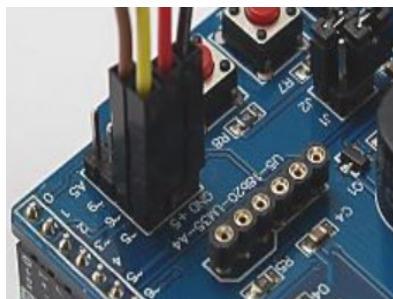
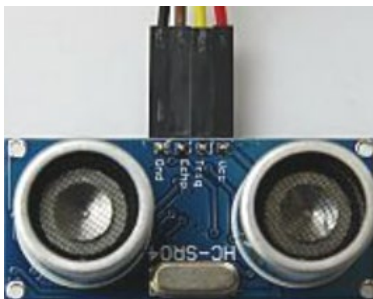
void setup() {
  Serial.begin(9600);
  MFS.initialize(&Timer1);
}

void loop() {
  int tempCentigrade = MFS.getLM35Data();
  MFS.write((float)tempCentigrade / 10, 1);
  delay(100);
}

```

Exemplo 7 – Usar o ultrassom HC SR04

Os pinos de disparo(Trigger) e eco(Eco) do módulo são ligados aos pinos 5 e 6 do Arduino, respetivamente.



```

#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>
const int TrigPin = 5;
const int EchoPin = 6;

void setup() {
  Serial.begin(9600);
  pinMode(TrigPin, OUTPUT);
}

```

```

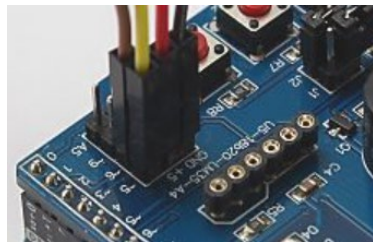
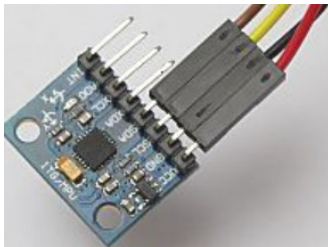
pinMode(EchoPin, INPUT);
Timer1.initialize();
MFS.initialize(&Timer1);
MFS.initSonar(SMOOTHING_MODERATE);
}

void loop() {
MFS.write((int)MFS.getSonarDataCm(TrigPin, EchoPin));
delay(100);
}

```

Exemplo 8 – Usar o sensor de movimento MPU6050

A placa usa a biblioteca Wire (protocolo I2C) para ler valores do sensor MPU6050. Pressione o botão 1 para alternar dados do sensor ou pressione o botão 2 para percorrer os valores de aceleração, giroscópio e temperatura.



```

#include <Wire.h>
#include <MultiFuncShield.h>
#include "I2C.h"
#include "MPU6050.h"

MPU6050 MPU;
void displayHeading(byte mode);

void setup() {
  Serial.begin(9600);
  Wire.begin();
  I2C1.initialize(&Wire);
  MPU.initialize(&I2C1, MPU_DEFAULT_ADDRESS);
}

```

```

MFS.initialize();
MFS.write("Acc");
}

byte displayValues = true;
byte displayMode = 0;

void loop() {
  byte btn = MFS.getButton();

  if (btn == BUTTON_1_PRESSED)
  {
    displayValues = !displayValues;

    if (displayValues)
    {
      displayHeading(displayMode);
    }
    else
    {
      MFS.write("Off");
    }
  }

  if (displayValues)
  {
    // Use button 2 to cycle though the display modes.
    if (btn == BUTTON_2_PRESSED)
    {
      displayMode++;

      if (displayMode == 3)
      {
        displayMode = 0;
      }

      displayHeading(displayMode);
    }
  }
}

```

```

if (displayMode == 0)
{
    // display raw acceleration values.
    MPU.getAccelRaw();
    Serial.print((float)MPU.accel_X_Raw / MPU.accelScaleValue);
    Serial.print("\t");
    Serial.print((float)MPU.accel_Y_Raw / MPU.accelScaleValue);
    Serial.print("\t");
    Serial.print((float)MPU.accel_Z_Raw / MPU.accelScaleValue);
    Serial.print("\t\n");
}
else if (displayMode == 1)
{
    // display raw gyrovalues
    MPU.getGyroRaw();
    Serial.print((float)MPU.gyro_X_Raw / MPU.gyroScaleValue);
    Serial.print("\t");
    Serial.print((float)MPU.gyro_Y_Raw / MPU.gyroScaleValue);
    Serial.print("\t");
    Serial.print((float)MPU.gyro_Z_Raw / MPU.gyroScaleValue);
    Serial.print("\t\n");
}
else if (displayMode == 2)
{
    Serial.println((float)MPU.getTemp10th() / 10);
}
}

delay(50);
}

void displayHeading(byte mode)
{
    if (mode == 0)
    {
        Serial.println("Acceleration in g (1g = 9.8 m/s/s)");
        Serial.println("X\tY\tZ");
        MFS.write("Acc");
    }
    else if (mode == 1)

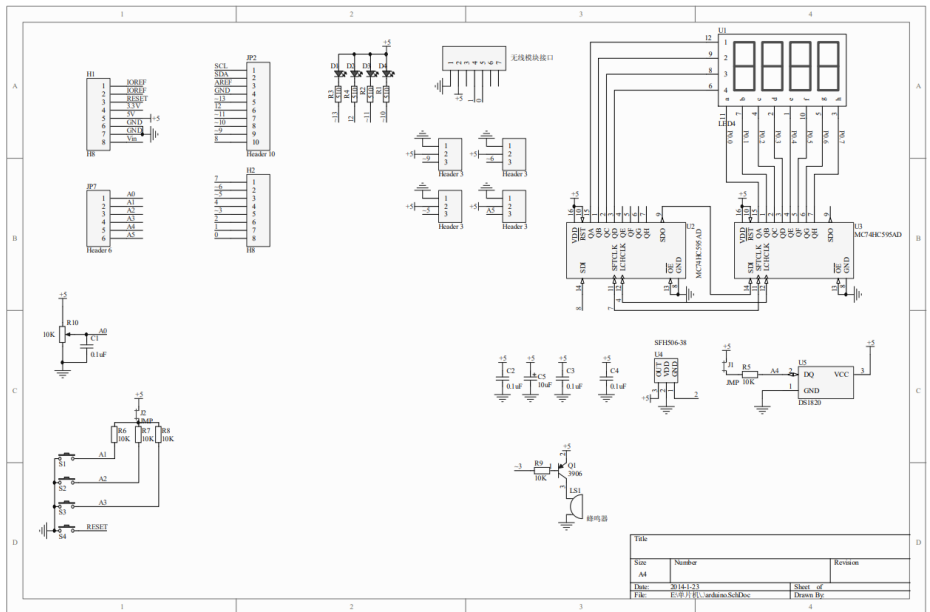
```

```

{
Serial.println("Gyro angular velocity in degrees / second");
Serial.println("X\tY\tZ");
MFS.write("Gyro");
}
else if (mode == 2)
{
Serial.println("Temperature in degrees celsius.");
MFS.write("Te");
}
}

```

5. Diagrama do circuito



Title		
Site	Number	Revision
A4		
Date	2014-5-31	Sheet of
File	C:\07\11\landrus\Sch\Doc	Drawn by

CE Declaration of Conformity

Company: Shenzhen Scope Corporation, Ltd.

Address: 12/13 Floors, C2 Building, I Park, No. 1001, College Road, Nanshan, Shenzhen, Guangdong, China

Product Name: Arduino compatible multifunction expansion board

Product Model: XUGU-E014 (AL.AZ00569)

Directives and Standard applicable:

Our samples match EMC directive 2014/30/EU and ROHS EU2011/65+AMD EU 2015/863

EMC Test Standards :EN 55032:2015+A11:2020+A1:2020

EN 55035:2017+A11:2020

RoHS Test Standards : EN 62321-5:2014,

EN62321-4:2014+A1:2017,

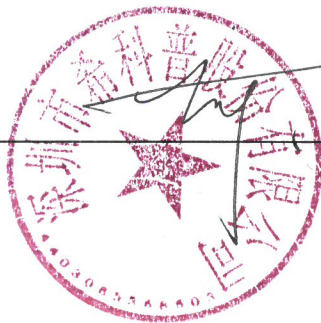
EN 62321-7-1:2015

EN 62321-7-2:2017,

IEC 62321-6:2015,

IEC 62321-8:2017

Signature: _____



Date: ___Oct. 11,2023___