



Sistema Solar em escala realista com *display* gráfico e Arduino Uno

.....
Camila Muniz de Oliveira¹
Edson Ribeiro de Britto de Almeida Junior^{1,*}

Néryla Vayne Justino Alves²
Ronaldo Celso Viscovini²

¹Programa de Pós-Graduação em Educação para a Ciência e a Matemática, Universidade Estadual de Maringá, Maringá, PR, Brasil.

²Departamento de Ciências, Universidade Estadual de Maringá, Campus Regional de Goioerê, Goioerê, PR, Brasil.

RESUMO:

O documento da Base Nacional Comum Curricular sinaliza que a maioria dos estudantes se interessa por objetos celestes. A importância desse assunto, para contribuir com as aprendizagens essenciais, comunga com as orientações da educação básica, que concebe o tema da astronomia para o desenvolvimento de competências específicas. Em contraposição à importância dada ao assunto, pesquisas demonstram certa fragilidade de encaminhamento metodológico do ensino de astronomia no contexto educacional brasileiro. Nesse sentido, o presente estudo objetiva contribuir para o desenvolvimento de uma prática alternativa para o ensino de conceitos inerentes à astronomia, especificamente no que diz respeito à compreensão das escalas de distância astronômicas entre os elementos existentes no Sistema Solar. Para isso, desenvolveu-se um Modelo do Sistema Solar em Escala Realista com o *display* gráfico conectado a um Arduino Uno. O propósito disso é simular uma viagem pelo Sistema Solar, visitando seus astros (Sol e planetas) numa escala realista, a fim de proporcionar aos estudantes explicações e previsões sobre as dimensões e características dos corpos do Sistema Solar com o uso de dispositivos digitais.

Palavras-chave: Sistema Solar; ensino de astronomia; produto educacional

.....

*Autor de correspondência. E-mail: erbj13@gmail.com.

1. Introdução

Astronomia é uma área do conhecimento que tem familiaridade com profusos ramos da ciência, favorecendo a compreensão dos paradigmas vigentes no decorrer da história, responsáveis pelas explicações do funcionamento do Universo e suas implicações no desenvolvimento da ciência moderna [1, 2]. Com relação aos avanços teóricos, que subsidiam os conteúdos específicos da astronomia, Siemsen e Lorenzetti [3] consideram que há, nessa área, uma crescente sistematização de atividades direcionadas à compreensão do processo de aprendizagem. Para eles, existem muitas pesquisas que priorizam as interações que ocorrem em sala de aula.

A Base Nacional Comum Curricular aponta que, no estudo das Ciências da Natureza para o ensino fundamental, a unidade temática Terra e Universo busca “a compreensão de características da Terra, do Sol, da Lua e de outros corpos celestes - suas dimensões, composição, localizações, movimentos e forças que atuam entre eles” [4, p. 328].

Vechi e cols. [5] corroboram que o “Sistema Solar” é uma área de conhecimentos astronômicos que, em geral, desperta a atenção e curiosidade dos estudantes para compreendê-lo. Porém, os referidos autores ressaltam que “é difícil comparar as dimensões relativas do Sistema Solar, pois os planetas são dezenas a centenas de vezes menores que o Sol e são milhares de vezes menores que as suas órbitas planetárias” [5, p. 2].

Contribuindo com o debate, Da Rosa e cols. [6] acentuam outro aspecto significativo, que sinaliza fragilidades no ensino dos conceitos de astronomia. Para os autores, a dificuldade está na necessidade de abstração por parte dos

alunos, já que é um sistema conceitual que envolve distâncias, com elevada ordem de grandeza. Os autores ainda ressaltam que a escala astronômica é um conceito tão delicado de ser abordado que, em geral, se apresenta como uma grande dificuldade para a compreensão do nosso universo, especialmente do Sistema Solar.

Nesse sentido, as Tecnologias da Informação e Comunicação (TIC) podem ser aliadas daqueles professores que desejam aprimorar suas práticas de ensino. Corroborando para isso, Brown [7, p. 432] complementa que a “visualização, proporcionada pelo uso das tecnologias, pode favorecer a criação de modelos e o seu uso para a compreensão das relações entre estes e o mundo real”. Nesse sentido, os recursos digitais podem proporcionar condições para vivenciarmos virtualmente situações que se aproximam das reais.

Diante de tais reflexões e em atenção às produções teóricas relacionadas aos aspectos da astronomia e de seu significado para a vida dos homens, comungamos com Darroz e cols. [8] de que a exploração desse tema ainda é um desafio para os profissionais da educação. Nesse sentido, entre os inúmeros conceitos inerentes à astronomia, optamos, no presente trabalho, por explorar as escalas de distâncias astronômicas por meio do desenvolvimento de um Modelo do Sistema Solar em Escala Realista, com o *display* gráfico conectado a um Arduino Uno.

2. Modelo do Sistema Solar em Escala Realista - MSSER

O produto educacional MSSER é composto de 11 *displays* gráficos com Arduino Uno e LCD - para os oito planetas, alguns planetas-anões (como Plutão e Ceres) e o Sol. Essa divisão de repre-

sentar cada astro supracitado por um *display* gráfico unitário se justifica tendo em vista que, para montar um sistema multimídia que contenha todo o Sistema Solar, em escala realista, ele seria virtualmente inviável, pois requeria uma tela de dezenas de metros.

O Arduino Uno é uma placa micro controladora baseada no ATmega328P. Ela tem 14 pinos de entrada/saída digital, 6 entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB, um conector de energia, um conector ICSP e um botão de *Reset*. A placa de Arduino Uno foi ligada a um LCD de 2,5 polegadas e a um botão de pressão de contato momentâneo (*push pull*) no pino A5, conforme representado na Fig. 1.

Para a construção do MSSER, é necessário explicitar escalas de tamanhos e distâncias para posicionar, adequadamente, os dispositivos gráficos ao longo do estabelecimento de ensino. Tal escolha depende do ambiente em que se pretende realizar a montagem. No caso do modelo construído, cada *display* foi posicionado a uma distância em escala de 1 m para 100.000.000 km (1:10¹¹), conforme indicado na Tabela 1.

Utilizando a escala (1 mm para 100.000 km), é apresentado, na Tabela 2, o diâmetro (real e em escala) do Sol e dos planetas (e alguns planetas-anões). Note que a maioria dos diâmetros, em escala, são frações de milímetros. Porém, como cada pixel do LCD tem tamanho de aproximadamente 150 µm, eles serão representados graficamente como um único pixel (≈150 µm) de cor branca, para qualquer apresentação.

Outros corpos solares relevantes são os satélites (luas), que, às vezes, são até maiores que alguns planetas e planetas-anões. Esses corpos orbitam astros celestes diferentes do Sol. É importante salientar que o número de satélites planetários utilizados neste projeto é parcial. Na Tabela 3, são apresentados os principais satélites do Sistema Solar (com diâmetro maior que 1.000 km), com seus diâmetros (reais) e raios orbitais (reais e em escala).

Cada dispositivo foi programado com características específicas dos corpos celestes do Sistema Solar, por exemplo: raio da órbita; composição; período de rotação e translação; informações de seus satélites (caso tenham), entre outras. A Fig. 2 agrupa imagens de tela contendo informações que foram utilizadas em alguns *displays*.

Para que isso seja possível, um cartão de memória contendo as imagens das informações de cada astro, assim

como a programação que será descrita adiante, deverá ser inserido na *case* da placa LCD. Cada imagem deverá ser salva no cartão de memória no formato *Windows Bitmap* (.bmp) e ter tamanho fixo de 320 x 240 pixels. Para economizar espaço e reduzir o tempo de leitura, as imagens devem ser gravadas com matriz de 16 bits para cor, no formato R5G6B5, ou seja, 5 bits para o vermelho (*Red5*), 6 bits para o verde (*Green6*) e 5 bits para o azul (*Blue5*). Por fim, para disciplinar a apresentação das páginas, os arquivos .bmp devem ser gravados no sentido de cima para baixo (*Top to Bottom*).

Para converter e adequar arquivos de imagem, podem ser utilizados diversos programas, pagos ou gratuitos, instaláveis ou que podem ser acessados pela internet (on-line). Para os testes apresentados, foi utilizado o software on-line *convert2bmp*^[x],

que permite ajustar o tamanho, a matriz de cor e a direção de apresentação. A elaboração das informações foi fundamentada no livro *Sistema Solar: uma exploração visual dos planetas, das luas e de outros corpos celestes que orbitam nosso Sol*, de Chown [9], e em conceitos do site oficial da Nasa. Para facilitar a



Figura 1 - Arduino Uno com *display* LCD mostrando a imagem de Saturno, com ampliação aparente de 10x.

Tabela 1: Planetas e alguns planetas-anões do Sistema Solar com a distância até o Sol (semieixo maior).

Planeta (* <i>planetas-anões</i>)	Distância do Sol	
	Real (km)	Escala (m)
Mercúrio	57.909.050	0,58
Vênus	108.208.930	1,08
Terra	149.598.261	1,50
Marte	227.939.100	2,28
* <i>Ceres</i>	414.010.000	4,14
Júpiter	778 547 200	7,79
Saturno	1.433.449.370	14,33
Urano	2.876.679.082	28,77
Netuno	4.503.443.661	45,03
* <i>Plutão</i>	5.906.376.272	59,06

Tabela 2: Diâmetro do Sol, dos planetas e de alguns planetas-anões do Sistema Solar.

Sol/planeta(* <i>planetas-anões</i>)	Diâmetro equatorial	
	Real (km)	Escala (mm)
Sol	1.391.400	13,91
Mercúrio	4.879	0,05
Vênus	12.103	0,12
Terra	12.756	0,13
Marte	6.792	0,08
* <i>Ceres</i>	945	0,01
Júpiter	142.984	1,43
Saturno	120.536	1,21
Urano	51.118	0,51
Netuno	49.528	0,50
* <i>Plutão</i>	2.374	0,02

Tabela 3: Principais satélites do Sistema Solar (diâmetro maior que 1.000 km).

Satélite (planeta)	Diâmetro real (km)	Raio orbital	
		Real (km)	Escala (mm)
Planeta Terra			
Lua	3.474	384.399	3,84
Planeta Júpiter			
Io	3.630	421.700	4,22
Europa	3.121	671.034	6,71
Ganimesdes	5.262	1.070.412	10,70
Calisto	4.820	1.882.709	18,83
Planeta Saturno			
Tétis	1.062	294.619	2,95
Dione	1.123	377.396	3,77
Reia	1.527	527.108	5,27
Titã	5.150	1.221.870	12,22
Jápeto	1.470	3.560.820	35,60
Planeta Urano			
Ariel	1.353	191.020	1,91
Umbriel	1.169	266.300	2,66
Titânia	1.576	435.910	4,36
Oberon	1.522	583.520	5,84
Planeta Netuno			
Tritão	2.705	354.759	3,55
Planetoide Plutão			
Caronte	1.207	19.571	0,20

reprodução do MSSER em outras instituições de ensino, a pasta com as imagens de informações específicas de cada astro pode ser baixada gratuitamente do endereço do autor.

Para controlar o *display*, foi utilizada a biblioteca Adafruit e, para controlar o conteúdo e a sequência da apresentação das imagens, deve haver no cartão de memória um arquivo nominado: *PAGES.cfg*. É um arquivo de texto simples que pode ser criado e editado por um bloco de notas (*Notepad*). Cada linha do arquivo deve conter no mínimo um número da página (n) seguido pelo símbolo igual (=) e pelo nome do arquivo de imagem: n = image.bmp.

Quando se pretende apresentar satélites (até o número de cinco) na ima-

gem, é necessário informar os respectivos raios orbitais ($R_1, R_2, R_3...$) em unidades de pixels, e seus períodos orbitais ($T_1, T_2, T_3...$): n = image.bmp; $R_1, T_1; R_2, T_2; R_3, T_3$.

A programação do MSSER foi desenvolvida no software Arduino, que é uma plataforma eletrônica de código aberto baseada em hardware e software, que não exige conhecimento exímio em linguagem de programação. A listagem da programação do MSSER é apresentada na íntegra no Material Suplementar, possibilitando que pessoas interessadas possam replicar esses dispositivos.

A programação prevê que, se o tempo de pressão do botão (*push pull*) for menor que um segundo, será passa-

do para a próxima página/imagem. Se este tempo for entre um e três segundos, é selecionada a página/imagem anterior e, se o tempo for maior que três segundos, é selecionada a primeira página/imagem. Assim, à medida que o aluno for clicando no botão existente no dispositivo, as informações serão apresentadas, simultâneas às imagens elaboradas em escala.

Como mencionado anteriormente, cada pixel do LCD tem tamanho de aproximadamente 150 μm . Consequentemente, em ampliações maiores que 3X, nos maiores satélites (Ganimesdes, Calisto e Titã), esta limitação pode causar uma distorção de tamanho relativo. Entretanto, conforme a Tabela 4, as órbitas destes satélites maiores não são visíveis no *display* LCD para ampliações iguais a 5X.

Com base na tabela supracitada, quando o valor de ampliação 1X resulta em zero, cada referido astro é representado por um pixel único de cor escura. É importante salientar que isso ocorre devido à utilização de uma escala única para descrever tanto as dimensões dos corpos do Sistema Solar como seus raios orbitais. O mesmo ocorre com as imagens referentes aos principais satélites do Sistema Solar. No entanto, em uma situação real, não é isso que acontece. Afinal, Marte é sete vezes maior que Ceres. Por conta disso, as escalas de ampliação adotadas nas figuras são diferentes, conforme indica a Tabela 5.

As imagens de satélites são representadas por um único pixel girando em torno do seu planeta num determinado período. Para facilitar a visualização do movimento, a escala de tempo usada na simulação é de um segundo por dia, conforme ilustrado na Fig. 3. Dessa forma, a Lua terrestre terá um período de apenas 27,32 s para representar os 27,32 dias reais.

Conforme o *display* apresentado na



Figura 2 - Imagem .bmp das informações sobre a estrutura de Júpiter (à esquerda). Imagem .bmp das informações sobre a superfície de Europa, satélite natural de Júpiter (à direita).

Tabela 4: Diâmetro (em pixels) dos principais astros do Sistema Solar, com quatro diferentes escalas de ampliação.

Astro	Escala de ampliação			
	1X	2X	5X	10X
Sol	93	186	>320	>320
Mercúrio	0	1	2	3
Vênus	1	2	4	8
Terra	1	2	4	9
Marte	0	1	2	5
Ceres	0	0	0	1
Júpiter	10	19	48	95
Saturno	8	16	40	80
Urano	3	7	17	34
Netuno	3	7	17	33
Plutão	0	0	1	2

Tabela 5: Raios das órbitas (em pixels) dos principais satélites do Sistema Solar, com diferentes escalas de ampliação.

Satélite	Período (dias)	Escala de ampliação		
		1X	2X	5X
Lua	27,32	26	51	128
Io	1,77	28	56	141
Europa	3,55	45	89	224
Ganimedes	7,15	71	143	357
Calisto	16,69	126	251	628
Tétis	1,89	20	39	98
Dione	2,74	25	50	126
Reia	4,52	35	70	176
Titã	15,94	81	163	407
Jápeto	79,33	237	475	1187
Ariel	2,52	13	25	64
Umbriel	4,14	18	36	89
Titânia	8,71	29	58	145
Oberon	13,46	39	78	195
Tritão	5,88	24	47	118
Caronte	6,39	1	3	7

Figura anterior, os demais também foram colocados em caixas de PVC de 4" X 2" (FAME) próprias para uso com cauletas, que são bastante utilizadas em instalações elétricas. As tampas foram usinadas e o material também foi feito de PVC para encaixar nos *displays* e no botão. A alimentação da placa Arduino foi uma fonte de 5 V por 2 A. Os seis primeiros Arduínos (Fig. 4) foram alimentados com uma única fonte e os cinco últimos, por estarem com maior distância entre eles, contaram com uma fonte para cada placa.

3. Considerações finais

Os recursos tecnológicos estão recebendo mais atenção na área da educação, conforme apresentado nos documentos oficiais. Nesse sentido, torna-se fundamental a compreensão das possi-

bilidades e das limitações de cada recurso tecnológico educacional com o intuito de refletir a respeito dos subsídios básicos para o desenvolvimento de atividades, sejam elas experimentais ou não, sobre as mais diversas práticas de ensino de ciências.

Entre as várias possibilidades de uso de recursos digitais para ensino de astronomia, optamos pelo desenvolvimento de um Modelo do Sistema Solar em Escala Realista (MSSER), com o *display* gráfico conectado a um Arduino Uno. Tal escolha



Figura 3 - Imagem de Saturno e de seus satélites naturais girando em um determinado período, ampliada em 2X.

se justifica por acreditarmos que este recurso pode auxiliar no processo de divulgação e de popularização da astronomia, tendo em vista que a sua utilização e instalação nas escolas não necessita de telescópios nem de outros instrumentos superequipados para o estudo do Sistema Solar.

As imagens com diferentes escalas de ampliação são elementos essenciais para que o estudante possa visualizar, virtualmente, a diferença das dimensões dos astros em escala real, o que não é possível a olho nu. Nesse sentido, o MSSER permite que os professores desenvolvam atividades de ensino que podem aproximar os estudantes do conhecimento científico, levando-os a vivenciar, virtualmente, situações que se aproximam da realidade. Com pequenas alterações nas imagens a serem apresentadas, o MSSER pode ser adaptado a outros assuntos ou áreas do conhecimento, como biologia (células, bactérias e vírus) e química (átomos e moléculas).

Recebido em: 13 de Janeiro de 2022

Aceito em: 16 de Fevereiro de 2022



Figura 4 - MSSER montado em um colégio privado de um município da mesorregião centro-ocidental paranaense.

Material Suplementar

MS1: Código de programação para o Sistema Solar em escala realística.

Referências

- [1] R. Caniato, *Um Projeto Brasileiro para o Ensino de Física*. Tese de Doutorado, Faculdade de Filosofia, Ciências e Letras de Rio Claro, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Rio Claro, 1973.
- [2] V. Queiroz, *A Astronomia Presente nas Séries Iniciais do Ensino Fundamental das Escolas Municipais de Londrina*. Tese de Doutorado, Mestrado em Ensino de Ciências e Educação Matemática, Universidade Estadual de Londrina, Londrina, 2008.
- [3] G.H. Siemsen, L.A. Lorenzetti, *ACTIO Docência em Ciências* **2**, 185 (2017).
- [4] Brasil, Ministério da Educação. *Base Nacional Comum Curricular - BNCC, 2ª versão*. (MEC/SemTec, Brasília, 2018).
- [5] A. Vechi, A.F. Brito, D.B. Valentim, M.E. Gozzi, A.R. Sampaio e cols., *Revista Brasileira de Ensino de Física* **35**, 2505 (2013).
- [6] Á.B. Da Rosa, A.C. Giacomell, C.T.W. Da Rosa, *Revista Ibero-Americana de Educação* **72**, 107 (2014).
- [7] J.P. Brown, *Táticas de Visualização para Resolver Tarefas do Mundo Real* (Universidade Federal de Ouro Preto, Ouro Preto, 2011).
- [8] L.M. Darroz, C.W. Da Rosa, A.B. Da Rosa, C.A.S. Pérez, *Revista Latino-Americana de Educação em Astronomia* **17**, 107, 2014.
- [9] M. Chown, *Sistema Solar: Uma Exploração Visual dos Planetas, das Luas e de Outros Corpos Celestes que Orbitam Nosso Sol* (Blucher, São Paulo, 2011).

Endereços de internet

https://drive.google.com/drive/folders/11H1xb_D9h7xQye-rl3CX6FkSM5h1yuJm?usp=sharing.

Material Suplementar

Programação do Sistema Solar em escala realística

```
#include "Adafruit_GFX.h"
#include "Adafruit_TFTLCD.h"
#include <SD.h>
#include <SPI.h>

String const
    cfgFILE="PAGES.CFG";

#define WHITE  0xFFFF
#define BLACK  0x0000
#define KEY    A5

#define CENTER_X 160
#define CENTER_Y 120

#define MAX_LUNES 5
struct strLune {
    boolean Active;
    int Radius;
    long Period;
    int LastX;
    int LastY;
    int LastColor;
};
strLune Lunes[MAX_LUNES];
strLune newLunes[MAX_LUNES];

struct {
    uint16_t ID;
    uint32_t Size;
    uint32_t Reserved;
    uint32_t Step;
} bmpHead;

Adafruit_TFTLCD
    tft(A3, A2, A1, A0, A4);

File cfgFile;
File bmpFile;

void setup() {
    //Serial.begin(9600);
    pinMode(KEY,INPUT_PULLUP);
    tft.reset();
    delay(500);
    tft.begin(tft.readID());
    tft.fillScreen(BLACK);
    tft.setRotation(1);
    // Testa SD Card
    if (!SD.begin(10)) {
        tft.setTextSize(4);
        tft.setCursor(12,100);
        tft.print("ERRO SD CARD");
        while(1); } // Erro SD Card
    // Testa Arquivo Configuração
    cfgFile =
        SD.open(cfgFILE,FILE_READ);
```

```

if (!cfgFile) {
    tft.setTextSize(4);
    tft.setCursor(12,80);
    tft.print("ERRO ARQUIVO");
    tft.setCursor(12,120);
    tft.print(cfgFILE);
    while(1); } // Erro Arquivo
}

int nowPage = 0;
int nextPage = 1;
boolean keyDown = false;
unsigned long timeKey = 0;
void loop() {
    unsigned long nowTime=millis();
    if (nowPage!=nextPage) {
        tft.drawPixel(0,0,BLACK);
        if (bmpDraw(getLunes(
            cfgPage(nextPage)))) {
            nowPage = nextPage;
            memcpy(Lunes,newLunes,
                sizeof(Lunes)); } }
    if (keyDown) {
        if (digitalRead(A5)) {
            keyDown = false;
            int tmp=nowTime-timeKey;
            if (tmp<1000)
                nextPage = nowPage+1;
            else if (tmp<3000)
                nextPage = nowPage-1;
            else
                nextPage = 1; } }
    else {
        if (!digitalRead(A5)) {
            keyDown = true;
            timeKey = nowTime; } }
    for (byte b=0;b<MAX_LUNES;++b){
        if ((Lunes[b].Radius!=0)&
            (Lunes[b].Period!=0)){
            long iAngle = nowTime %
                Lunes[b].Period;
            float fAngle=6.28*iAngle /
                Lunes[b].Period;
            tft.drawPixel(
                Lunes[b].LastX,
                Lunes[b].LastY,
                Lunes[b].LastColor);
            Lunes[b].LastX=CENTER_X+
                Lunes[b].Radius*
                cos(fAngle);
            Lunes[b].LastY=CENTER_Y+
                Lunes[b].Radius*
                sin(fAngle);
            Lunes[b].LastColor=
                tft.readPixel(
                    Lunes[b].LastX,
                    Lunes[b].LastY);
            tft.drawPixel(
                Lunes[b].LastX,
                Lunes[b].LastY,
                WHITE); } }
    // Espera para Atualizar

```

```

delay(25);
}

String cfgPage(int page) {
    cfgFile.seek(0);
    while (cfgFile.available()) {
        String sTemp =
            cfgFile.readStringUntil
                ('\r');
        sTemp.replace(" ", "");
        int8_t bPos=
            sTemp.indexOf('=');
        if (bPos>0) {
            if (page==sTemp.
                substring(0,bPos).
                toInt())
                return sTemp.
                    substring(bPos+1); } }
    return NULL; }

String getLunes(String sLine) {
    int8_t bImage =
        sLine.indexOf(';');
    if (bImage>0) {
        String sTemp =
            sLine.substring(bImage+1);
        for (byte b=0;
            b<MAX_LUNES; ++b){
            int8_t bMid =
                sTemp.indexOf(',');
            int8_t bEnd =
                sTemp.indexOf(';');
            newLunes[b].Radius=
                sTemp.substring(0,bMid)
                    .toInt();
            if (bEnd>0) {
                newLunes[b].Period =
                    long(1000*sTemp.
                        substring(bMid+1,bEnd).
                            toFloat());
                sTemp=sTemp.substring(
                    bEnd+1); }
            else {
                newLunes[b].Period =
                    long(1000*
                        sTemp.substring(bMid+1).
                            toFloat());
                sTemp=""; }
            newLunes[b].LastX=-1;
            newLunes[b].LastY=-1; }
        return sLine.substring(0,
            bImage); }
    else {
        newLunes[0].Radius=0;
        newLunes[1].Radius=0;
        newLunes[2].Radius=0;
        newLunes[3].Radius=0;
        newLunes[4].Radius=0;
        return sLine; } }

```

```

// ROTINA DE LEITURA DE IMAGEM
#define BUFFER_SIZE 160

```



```
uint16_t bufferData[BUFFER_SIZE];
boolean first;
boolean bmpDraw(String filename) {
  if (bmpFile=SD.open(filename.
    c_str(),FILE_READ)){
    bmpFile.read(&bmpHead,
      sizeof(bmpHead));
    bmpFile.seek(bmpHead.Step);
    uint16_t NumBlock =
      (bmpHead.Size-bmpHead.Step)
        / (2*BUFFER_SIZE);
    first = true;
    for (int i=0;i<NumBlock;++i) {
      bmpFile.read(bufferData,
        2*BUFFER_SIZE);
      tft.pushColors(bufferData,
        BUFFER_SIZE, first);
      first=false; }
    bmpFile.close();
    return true; }
  bmpFile.close();
return false; }
```