COMPONENTES DO SISTEMA

1 caixa suporte de plástico rígido com tampa
1 SmartNode 3S com processador SAMD51 (versão 6)
2 FT Click para conexão de fios
1 FT Click Solar para conexão do painel fotovoltaico
1 Display LCD 16x2 com backlight azul e adaptador I2C
1 teclado matricial 4x4 rígido preto
1 conector usb externo
2 conectores de dados externo
1 conector de energia externo
1 Fonte com cabo conector
1 motor WS23-0300-30-4
1 Encoder Omron E6B2-CWZ6C
1 Sensor de pressão 1.2MPa
1 Válvula TEZCA 5 setores

MANUAL DE UTILIZAÇÃO

Visão Geral

O sistema permite controlar a irrigação de setores agrícolas através da configuração do tempo de irrigação em cada setor. O usuário, através do teclado, pode escolher entre configurar o mesmo tempo para todos os setores ou tempos individuais para cada um. O sistema só entra em funcionamento quando a bomba é ligada, identificando esse estado através de um sensor de pressão de água.

Configuração inicial

Ao iniciar o sistema o usuário tem a possibilidade de escolher entre configurar o mesmo tempo para todos os setores ou configurar o tempo de cada setor individualmente. Portanto, deverá responder à pergunta “Mesmo tempo para os setores?”.

Setores com tempos iguais

Em caso afirmativo, o usuário deverá pressionar a tecla de confirmação ‘A’. Logo após será solicitado o tempo em que se deseja que os setores fiquem ligados, o usuário então insere o tempo, em MINUTOS, limitado a três dígitos e então pressiona a tecla de confirmação ‘A’.

Setores com tempos diferentes

Caso o usuário deseje que o tempo para cada setor seja diferente, deverá responder à pergunta inicial “Mesmo tempo para os setores?” com a tecla ‘B’, para negar a pergunta. Logo após será solicitado, para cada setor, o tempo que se deseja que o respectivo setor fique ligado, o usuário então insere o tempo, em MINUTOS, limitado a três dígitos e então pressiona a tecla de confirmação ‘A’, após cada inserção de tempo.

Sensor de Pressão

O sensor de pressão está configurado para a unidade em MPa, que é cerca de 9,87 atm. Quando é identificada pressão acima de 0,3 MPa(2,96 Atm) o sistema é acionado automaticamente. Esse valor pode ser facilmente alterado via código alterando a função pressão().

Reconfiguração do Sistema

A reconfiguração é realizada com o sistema desligado. Portanto, caso o sistema esteja em funcionamento, deve-se pressionar a tecla ‘D’ para desligá-lo. Com o sistema desligado pressione a tecla ‘C’ para iniciar o processo de reconfiguração do sistema, como foi feito na configuração inicial.

PROGRAMAÇÃO

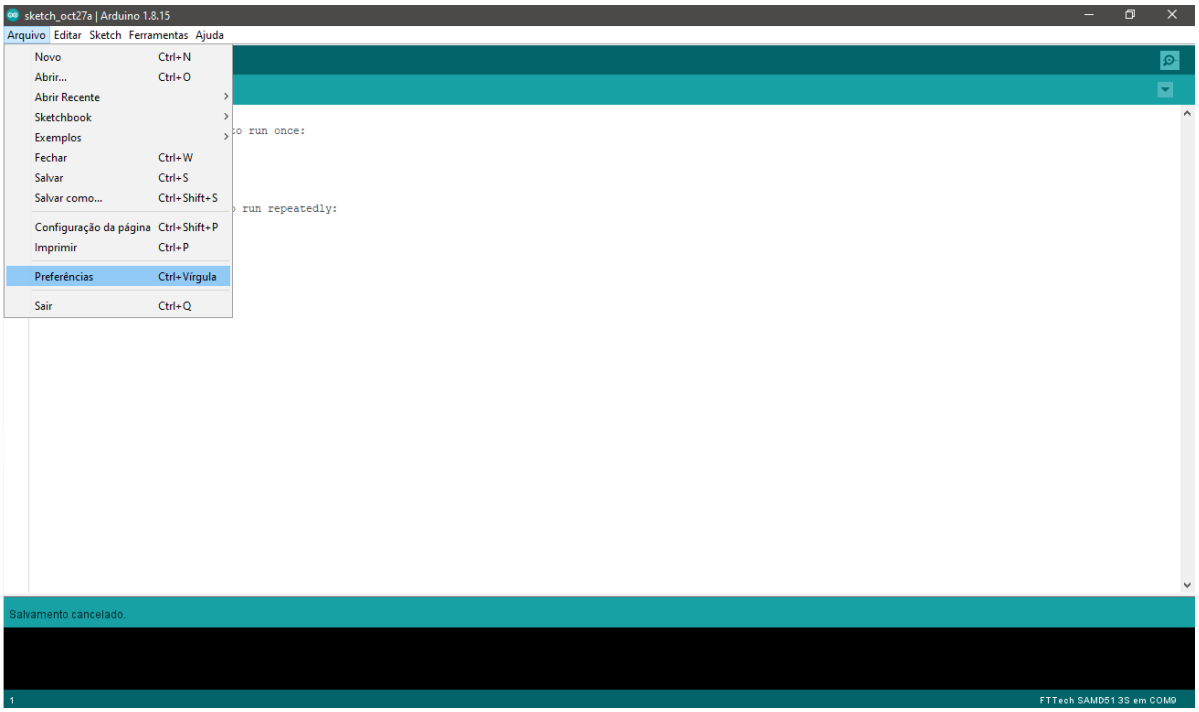
Versão da Placa e Versões das Bibliotecas

A caixa está equipada com o microcontrolador FTTech SmartNode 3S Versão 6. Para realizar alterações no programa o passo a passo a seguir deverá ser seguido.

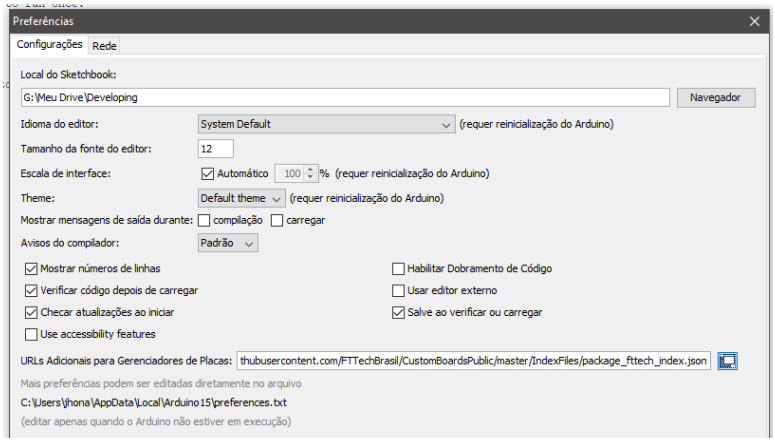
1 - Realize o download da versão 1.8 da Arduino IDE em https://www.arduino.cc/en/software

2 - Configure a IDE para descoberta da placa

- Abra o menu **Arquivo > Preferências**

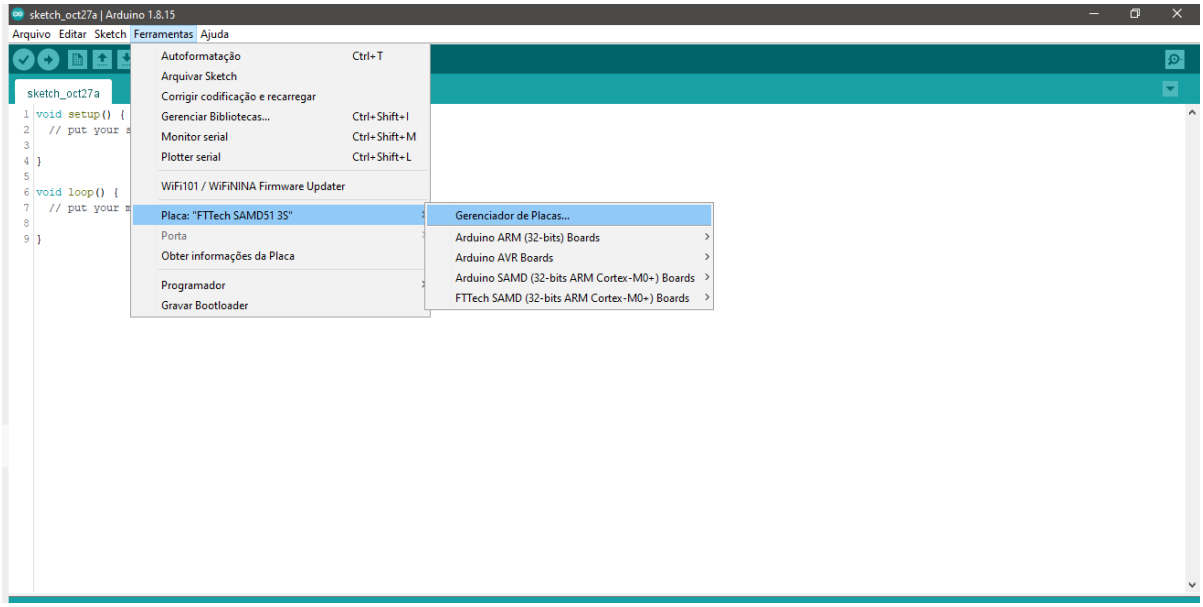


- No campo "**URLs adicionais para gerenciadores de placa**" insira a seguinte URL: https://raw.githubusercontent.com/FTTechBrasil/CustomBoardsPublic/master/IndexFiles/package_fttech_index.json e clique em OK:

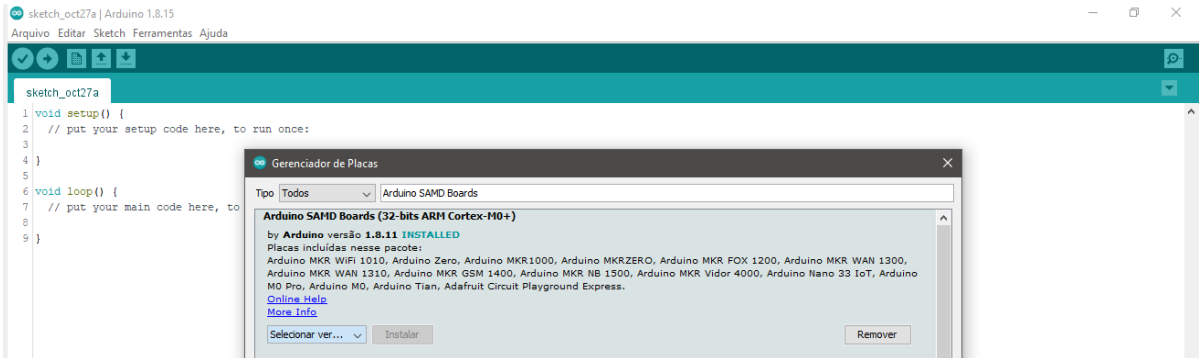


- Agora o Arduíno está habilitado para reconhecer a placa. Feche e abra novamente a IDE para que as configurações sejam efetivadas.

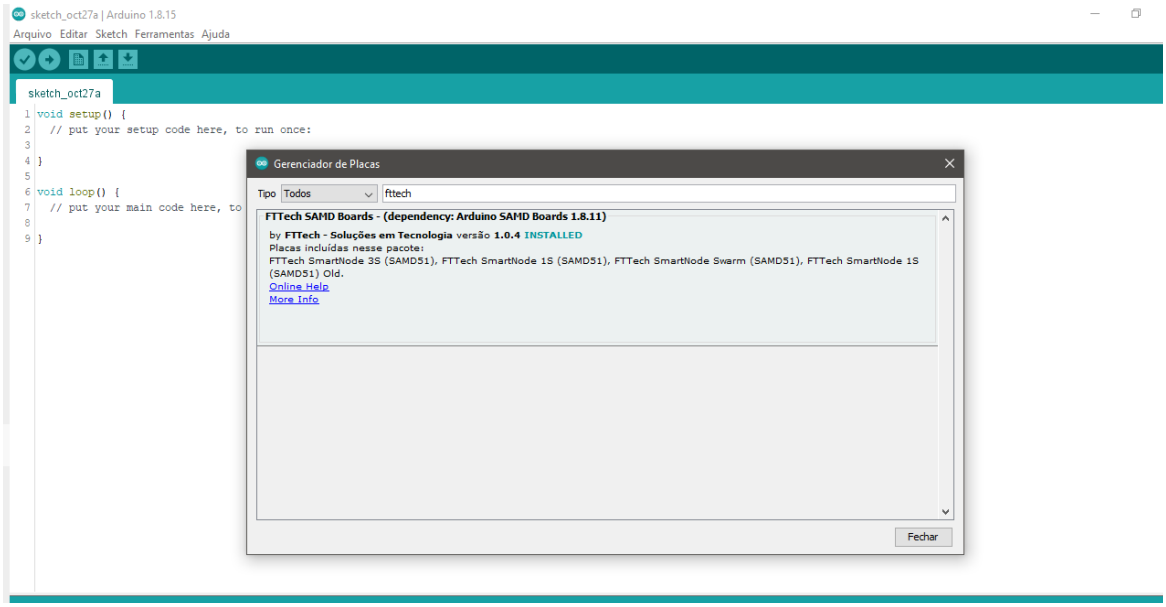
3 - Antes de prosseguir com a instalação da placa na IDE é necessário instalar uma dependência. Abra o menu **Ferramentas > Placa: > Gerenciador de Placas...**



- Com o Gerenciador de Placas aberto busque por “**Arduino SAMD Boards**” e instale a placa “**Arduino SAMD Boards (32-bits ARM Cortex-M0+)**” em sua **versão 1.8.11**


sketch_oct27a | Arduino 1.8.15
Arquivo Editar Sketch Ferramentas Ajuda
sketch_oct27a
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to
}
Gerenciador de Placas
Tipo Todos
Arduino SAMD Boards
Arduino SAMD Boards (32-bits ARM Cortex-M0+)
by Arduino versão 1.8.11 INSTALLED
Placas incluídas nesse pacote:
Arduino MKR WiFi 1010, Arduino Zero, Arduino MKR1000, Arduino MKRZERO, Arduino MKR FOX 1200, Arduino MKR WAN 1300, Arduino MKR WAN 1310, Arduino MKR GSM 1400, Arduino MKR NB 1500, Arduino MKR Vidor 4000, Arduino Nano 33 IoT, Arduino M0 Pro, Arduino M0, Arduino Tian, Adafruit Circuit Playground Express.
Online Help
More Info
Selecionar ver...
Instalar
Remover

4 - Agora já é possível instalar a placa. Busque por FTTech e instale a placa “**FTTech SAMD Boards (dependency: Arduino SAMD Boards 1.8.11)**” em sua **versão 1.0.4**



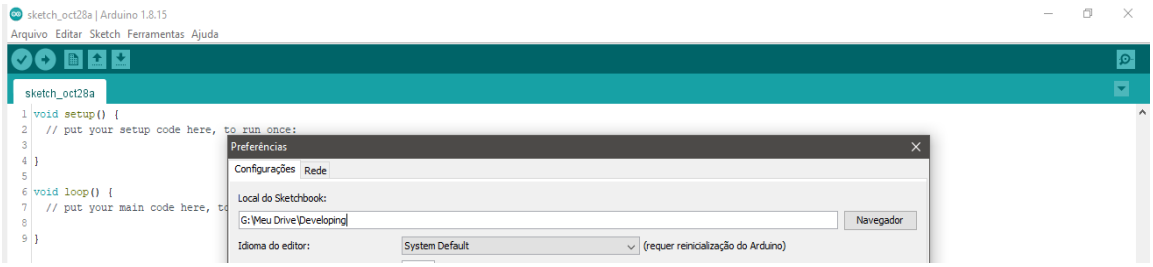
- Agora a placa já está devidamente instalada.

5 - Em seguida, as bibliotecas deverão ser instaladas:

- Abra o menu **Ferramentas > Gerenciador de Bibliotecas…**
- Busque por “**FTTech**” e instale a biblioteca “**FTTech SAMD51 Clicks**” na **versão 1.3.2**
- Busque por “**LiquidCrystal I2C**” e instale a biblioteca “**LiquidCrystal I2C**” na **versão 1.1.2**
- Busque por “**RTC SAMD51**” e instale a biblioteca “**Seeed Arduino RTC**” na **versão 2.0.0**
- Busque por “**SPIMemory**” e instale a biblioteca “**SPIMemory**” na **versão 3.4.0**

6 - Algumas bibliotecas não estão disponíveis no Gerenciador de Bibliotecas e devem ser instaladas manualmente.

- No menu **Arquivo > Preferências** verifique o caminho onde estão salvas as bibliotecas em “Local do Sketchbook”



- Na pasta “Libraries” salve as bibliotecas adicionais disponíveis em:

https://drive.google.com/file/d/1EcOD6GVBtsy3RYTllQowamyyKI0AoFRa/view?usp=sharing

https://drive.google.com/file/d/1JKE8Ie-xCsbwzEJMUGYq7_RLtut3dMgR/view?usp=sharing

DEFINES

DEBUG: Variável para depurar código. Quando recebe true exibe informações do sistema na porta Serial. Default: false

ENCODER_NUMBER_POS: Variável que define o tamanho do vetor “ENCODER_POSITION”, que guarda as posições relativas do encoder para posições nas válvulas e entre válvulas. Default: 10

DEBUG_BAUDRATE: Variável que define a velocidade da comunicação serial em bits por segundo. Default: 9600

TRANSITION_PRESSURE: Variável que define a pressão de transição que fará com que o sistema seja ativado. Default: 0.3

MOTOR_DIRECTION: Variável que define a direção de rotação do motor. Quando 0 a rotação ocorre para a direita, quando 1 a rotação ocorre para a esquerda. Default: 1

MOTOR_PULSE_DELAY: Variável que define o tempo de duração de um pulso do motor de passo em microssegundos. Cada pulso representa um passo do motor. Default: 1500

VALVE_TIME_SECTOR: Define o primeiro endereço de memória do setor onde são guardados os tempos em que cada válvula estará ativa. Default: 0

CURRENT_VALVE_ADRESS: Endereço de memória em que é guardado o número da válvula ativa. Default: 4096

TIME_LEFT_SECTOR: Define o primeiro endereço de memória do setor onde são guardados os tempos RESTANTES em que cada válvula deverá ficar ativa. Default: 8192

PRESSURE_SENSOR_PIN: Define o pino analógico onde está conectado o sensor de pressão. Default: A1

***** Para as variáveis que guardam endereços de memória o valor 255 significa que este endereço está vazio.***

FUNÇÕES

float fromPressureSensor_GetVoltage(void)
Definição: Realiza a leitura do sensor de pressão e converte o valor em tensão
Inputs: Nenhum
Output: A tensão em volts, tipo float
Pré-condição: Sensor de pressão conectado à porta analógica A1
Pós-condição: Sem alterações nas variáveis globais

float fromPressureSensor_GetMPaPressure(void)
Definição: Realiza a leitura do sensor de pressão e converte o valor lido em pressão
Inputs: Nenhum
Output: A pressão em MPa, tipo float
Pré-condição: Sensor de pressão conectado à porta analógica A1
Pós-condição: Sem alterações nas variáveis globais

bool pression(void)
Definição: Avalia se a pressão no sensor ultrapassa o limite definido pela variável "TRANSITION_PRESSURE"
Inputs: Nenhum
Output: Retorna **true** caso a pressão seja maior ou **false** caso seja menor ou igual
Pré-condição: Sensor de pressão conectado à porta analógica A1
Pós-condição: Sem alterações nas variáveis globais

void systemBegin(void)
Definição: Faz o setup inicial da placa e de todos os componentes do sistema e declara as interrupções
Inputs: Nenhum
Output: Nenhum
Pré-condição: Objetos de cada componente criados previamente (lcd, motor, encoder, rtc, memória flash e teclado)
Pós-condição: Setup inicial da placa e dos componentes e declaração das interrupções realizados com sucesso

void systemSetup(void)
Definição: Configura o sistema para a posição inicial, na primeira válvula. A partir desse ponto o motor e o encoder são resetados. Caso não tenha sido registrado na memória o número da válvula em operação, a função escreve a válvula 1 como válvula de operação.
Inputs: Nenhum
Output: Nenhum
Pré-condição: Objetos dos componente motor, encoder e memória flash criados previamente
Pós-condição: O sistema se encontra na primeira válvula e o endereço da válvula atual está preenchido.

void goToValv(uint8_t valv)
Definição: Configura o sistema para a posição inicial, na primeira válvula. A partir desse ponto o motor e o encoder são resetados. Caso não tenha sido registrado na memória o número da válvula em operação, a função escreve a válvula 1 como válvula de operação.
Inputs: Número da válvula a qual se deseja ativar
Output: Nenhum
Pré-condição: Número da válvula válido e objetos dos componente motor e encoder criados previamente
Pós-condição: O sistema se encontra na válvula “valv”

bool needConfig(SPIFlash flash)
Definição: Avalia se o sistema precisa de configuração de tempo para as válvulas
Inputs: Objeto do componente memória flash
Output: Retorna **true** caso seja necessário ou **false** caso não seja
Pré-condição: Objeto do componente memória flash válido e inicializado previamente
Pós-condição: Sem alterações nas variáveis globais

void resetSystem(void)
Definição: Reinicia o sistema e solicita nova configuração de tempo
Inputs: Nenhum
Output: Nenhum
Pré-condição: Objetos dos componente motor, encoder e memória flash válidos e inicializados previamente
Pós-condição: Sistema totalmente reiniciado

void memorySaveTime(uint16_t timing,SPIFlash flash, uint8_t valv, uint16_t firstAddress)
Definição: Escreve o tempo “timing” em dois endereços da memória flash de acordo com a válvula “valv” e o endereço “firstAddress”, primeiro endereço do setor de memória a ser salvo o tempo.
Inputs: o tempo que se deseja salvar, o objeto do componente memória flash, a válvula para qual se deseja salvar o tempo e o endereço da primeira posição da memória onde se deseja salvar o tempo (VALVE_TIME_SECTOR ou TIME_LEFT_SECTOR)
Output: Nenhum
Pré-condição: Objeto memória flash válido e inicializado previamente. Tempo, número da válvula e endereço "firstAddress" válidos (VALVE_TIME_SECTOR ou TIME_LEFT_SECTOR)
Pós-condição: Tempo salvo no setor de memória corretamente

uint16_t memoryGetTime(SPIFlash flash, uint8_t valv, uint16_t firstAddress)
Definição: Recupera o tempo salvo em dois endereços da memória flash de acordo com a válvula “valv” e o endereço “firstAddress”, primeiro endereço do setor de memória de onde será recuperado o tempo.
Inputs: O objeto do componente memória flash, a válvula para qual se deseja recuperar o tempo e o endereço da primeira posição da memória de onde se deseja recuperar o tempo.
Output: O tempo recuperado, tipo uint16_t
Pré-condição: Objeto do memória flash válido e inicializado previamente, número da válvula válido e endereço "firstAddress" válido (VALVE_TIME_SECTOR ou TIME_LEFT_SECTOR)
Pós-condição: Variáveis globais sem alteração

void setTime(void)
Definição: Configura o tempo em que as válvulas ficarão em cada setor. A função interage com o usuário e salva na memória os valores escolhidos.
Inputs: Nenhum
Output: Nenhum
Pré-condição: Objetos dos componentes memória flash e LCD criados e inicializados previamente. Objeto do componente teclado criado previamente
Pós-condição: Tempos escolhidos pelo usuário para cada válvula salvos na memória flash

void keepTime(uint16_t _time)
Definição: Mantém o sistema na válvula por “time” minutos
Inputs: O tempo em que a válvula ficará ativa
Output: Nenhum
Pré-condição: Objetos dos componentes memória flash e RTC criados e inicializados previamente. Objeto Datetime criado e inicializado previamente
Pós-condição: A função aguarda por “time” minutos antes de encerrar e passar para a próxima instrução do programa.