

Ponto de controle 4

Monitoramento Remoto

Lucas Luan A. Barbosa
Matrícula: 13/0122173
Universidade de Brasília
E-mail: lucasluan.fga@gmail.com

Jackson Paz
Matrícula: 13/0028789
Universidade de Brasília
E-mail: jackson.paz@gmail.com

Resumo— No meio rural é cada vez maior a pecuária de subsistência, esta consiste na criação de animais de grande porte, em principal bovinos, com finalidade para abate ou ordenha. Os animais com finalidade laticínio, não somente atende a família rural, bem como fornece material para pequenos laticínios, fomentando assim grandes cooperativas no interior do país. O sistema cooperativo com a finalidade para abastecimento de beneficiadores do leite movimenta a economia local, portanto, gerando renda aos moradores nas periferias do laticínio. O sistema de armazenamento para estes pequenos produtores, consiste em tanques de aço inox que são alocados em áreas estratégicas para abastecimento pelo produtor local, e que em períodos pré determinado são coletados pelo beneficiador. O processo de coleta ocorre de forma sistemática baseado em datas pré determinadas, ocorrendo inconstância no maior poder de armazenamento do líquido, pois por muitas vezes a demanda é variável, ocorrendo excesso ou escassez de produção e impossibilitando o transporte de maior volume lácteo possível. Este estudo visa apresentar uma solução simples e prática sendo ela o monitoramento dos reservatórios através de sensores de nível e temperatura via comunicação GSM, apresentando ao beneficiário um melhor controle de qualidade do produto a ser adquirido assim como um meio de identificar o nível correto para enviar caminhões coletores aproveitando melhor a sua capacidade.

Palavras-Chaves—*RaspBerryPI, automatização, produção de leite, monitoramento*

I. JUSTIFICATIVA

O grande aumento da produção de leite e seus derivados tem se destacado no cenário econômico nacional, sendo esta uma das principais fontes de renda em algumas regiões interioranas dos estados brasileiros.

Muitos produtores rurais investem em gado leiteiro e em tanques de expansão para armazenar o produto, no entanto, esta não é uma forma muito rentável de se trabalhar individualmente na produção em pequena escala, uma vez que existe uma grande diferença entre o preço recebido pelo pequeno e pelo grande produtor de leite por dois motivos: qualidade e volume.

Para amenizar essa desigualdade, alguns criadores de gado leiteiro resolveram mudar esta forma de trabalho montando cooperativas para aumentar a quantidade de leite

vendido, pois assim, a cooperativa funciona como um grande produtor e consequentemente o valor recebido pelo produto é maior quando comparado ao produto fornecido em pequena escala.

Geralmente os compradores de leite em grande escala são os laticínios que usam desta matéria prima para produzir seus derivados aumentando o valor agregado ao produto, sendo que o sistema de coleta do mesmo é feito em alguns casos de forma semanal, onde é esperado que o reservatório de armazenamento de leite esteja cheio para que sejam enviados caminhões específicos para transporte desta matéria prima, pois desta maneira o comprador também economiza com frete e garante que o caminhão voltará com o máximo de proveito por viagem. No entanto, muitas vezes acontece de os reservatórios encherem antes do prazo estipulado para coleta e o comprador não aparecer no dia, impossibilitando que o produtor armazene mais leite, gerando assim prejuízo tanto para o comprador quanto para o produtor que deixa de produzir por não haver espaço disponível para armazenamento.

II. OBJETIVOS

Este projeto tem como principal objetivo desenvolver um sistema para monitoramento e coleta, bem como, disponibilizar dados sobre armazenamento para a indústria de laticínios, que consiga detectar a porcentagem de armazenamento de leite em tanques e sua temperatura local. Assim como outros objetivos:

1. Estudar e analisar as formas estocagem de líquidos;
2. Estudar sensores de detecção do volume de líquidos;
3. Estudar sensores para monitoramento de temperatura;
4. Estudo do microprocessador ARM Cortex-A53 quad-core de 1.2 GHz 64 bits embarcado na RaspBerry PI3;
5. Testar e verificar o funcionamento do protótipo;

III. REQUISITOS

O protótipo de monitoramento remoto será constituído de alguns periféricos básicos, como sensor de nível, temperatura,

módulo 3G, conversor A/D e Raspberry Pi 3, que serão responsáveis por realizar o monitoramento do tanque reservatório de leite e enviar em tempo real dados de temperatura e nível para um servidor.

O protótipo realizará de início as seguintes funcionalidades básicas:

- Coletar dados de temperatura do reservatório;
- Coletar dados de nível do reservatório;
- Enviar os dados coletados do reservatório para um servidor;

IV. BENEFÍCIOS

Com a automatização do monitoramento dos tanques de leite, possibilita que os compradores consigam sempre mandar caminhões para coletar o produto toda vez que os tanques estiverem cheios, mantendo uma regularidade na coleta e certificando-se quanto a qualidade do produto.

V. VISÃO GERAL DO SISTEMA

Diagrama de Blocos

O diagrama mostrado abaixo descreve a comunicação dos periféricos com o processador Raspberry Pi.

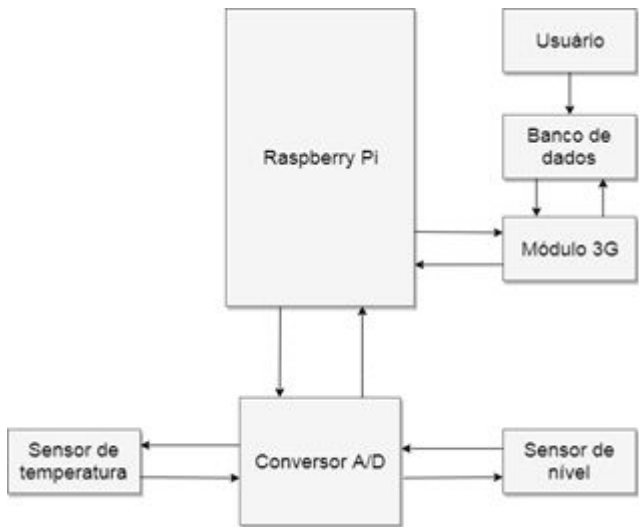


Figura 01: Diagrama de Blocos.

O usuário acessa o servidor para ter acesso aos dados do reservatório. A Raspberry solicita os dados de temperatura e nível aos sensores por meio de um conversor A/D e os envia ao servidor.

Fluxograma

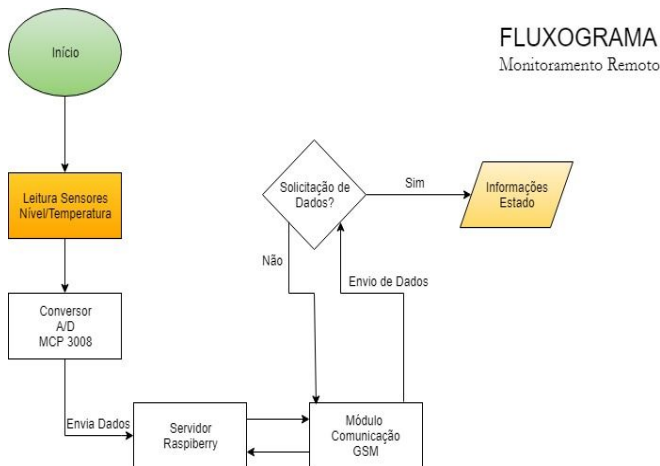


Figura 02: Fluxograma de funcionamento.

Periféricos

- Conversor A/D

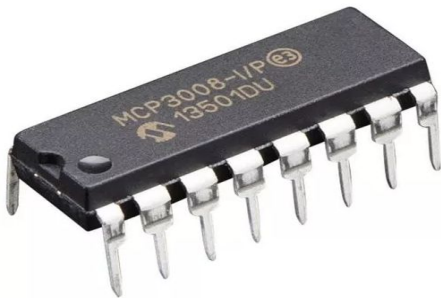


Figura 03: MCP3008.

Nome técnico: MCP3008.

Funcionalidade no projeto: Converter os sinais analógicos dos sensores para se comunicar com a Raspberry Pi.

Comunicação: Interface serial SPI.

- Sensor de nível



Figura 04: Higrômetro.

Nome técnico: Higrômetro.

Funcionalidade no projeto: Identificar o nível de volume no reservatório.

Comunicação: Serial analógica.

- Sensor de temperatura



Figura 05: Sensor de temperatura a prova de água DS18B20.

Nome técnico: DS18B20.

Funcionalidade no projeto: Coletar a temperatura do tanque de resfriamento.

Comunicação: Serial.

- Módulo 3G



Figura 06: Módulo GSM 3G.

Nome técnico: Módulo GSM 3G

Funcionalidade no projeto: Possibilitar o acesso remoto da Raspberry via rede GSM. Em síntese, o envio dos dados sensoriais via servidor em rede.

Comunicação: USB

- Processador.



Figura 07: Raspberry Pi 3B.

Nome técnico: Raspberry Pi 3B.

Funcionalidade no projeto: Executar as instruções necessárias para o projeto e gerir os periféricos.

Comunicação: I2C, SPI, Ethernet, HDMI.

" O'Reilly Media, Inc.", 2012.

- [3] Souza, M. R., M. M. O. P. Cerqueira, and M. J. Sena. "Avaliação da qualidade do leite resfriado, estocado em propriedades rurais por 48 horas e recebido por uma indústria de laticínios." *Congresso Nacional de Laticínios*. Vol. 16. 1999.
- [4] Augusto, M. E., Teixeira, M. A. S., Rosa, R. V., & de Sousa, A. L. (2017, October). Monitoramento de nível de rio utilizando microcomputador, sensor ultrassônico e comunicação celular. In *Congresso Nacional de Inovação e Tecnologia*.
- [5] de Oliveira, S. (2017). *Internet das Coisas com ESP8266, Arduino e Raspberry Pi*. Novatec Editora.

VI. REVISÃO BIBLIOGRÁFICA

- [1] Portal de notícias G1, Globo Rural. Disponível em: <http://g1.globo.com/economia/agronegocios/noticia/2012/09/sistema-d-e-cooperativa-beneficia-pequenos-produtores-de-leite-de-mg.html>>. Acesso em 2 de abril de 2018.
- [2] Richardson, Matt, and Shawn Wallace. *Getting started with raspberry PI*.

Anexos

- 1- Código usado para realizar a conversão A/D do sinal obtido pelo Higrômetro com o MCP3008:

```
#define _GNU_SOURCE

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdint.h>
#include <string.h>
#include <errno.h>
#include <wiringPi.h>
#include <wiringPiSPI.h>

#define TRUE (1==1)
#define FALSE (!TRUE)
#define CHAN_CONFIG_SINGLE 8
#define CHAN_CONFIG_DIFF 0

static int myFd ;

char *usage = "Usage: mcp3008 all|analogChannel[1-8] [-1] [-ce1] [-d]";

void loadSpiDriver()
{
    if (system("gpio load spi") == -1)
    {
        fprintf (stderr, "Can't load the SPI driver: %s\n", strerror (errno)) ;
        exit (EXIT_FAILURE) ;
    }
}

void spiSetup (int spiChannel)
{
    if ((myFd = wiringPiSPISetup (spiChannel, 1000000)) < 0)
    {
        fprintf (stderr, "Can't open the SPI bus: %s\n", strerror (errno)) ;
        exit (EXIT_FAILURE) ;
    }
}

int myAnalogRead(int spiChannel,int channelConfig,int analogChannel)
{
    if(analogChannel<0 || analogChannel>7)
        return -1;
    unsigned char buffer[3] = {1}; // start bit
    buffer[1] = (channelConfig+analogChannel) << 4;
    wiringPiSPIDataRW(spiChannel, buffer, 3);
    return ( (buffer[1] & 3 ) << 8 ) + buffer[2];
}

int main (int argc, char *argv [])
{
    int loadSpi=FALSE;
    int analogChannel=0;
    int spiChannel=0;
    int channelConfig=CHAN_CONFIG_SINGLE;
    if (argc < 2)
    {
        fprintf (stderr, "%s\n", usage) ;
        return 1 ;
    }
    if((strcasecmp (argv [1], "all") == 0) )
        argv[1] = "0";
    if ( ( sscanf (argv[1], "%i", &analogChannel)!=1) || analogChannel < 0 || analogChannel > 8 )
    {
        printf ("%s\n", usage) ;
        return 1 ;
    }
}
```

```

}
int i;
for(i=2; i<argc; i++)
{
    if (strcasecmp (argv [i], "-l") == 0 || strcasecmp (argv [i], "-load") == 0)
        loadSpi=TRUE;
    else if (strcasecmp (argv [i], "-ce1") == 0)
        spiChannel=1;
    else if (strcasecmp (argv [i], "-d") == 0 || strcasecmp (argv [i], "-diff") == 0)
        channelConfig=CHAN_CONFIG_DIFF;
}
//
if(loadSpi==TRUE)
    loadSpiDriver();
wiringPiSetup () ;
spiSetup(spiChannel);
//
if(analogChannel>0)
{
    printf("MCP3008(CE%d,%s): analogChannel %d = %d\n",spiChannel,(channelConfig==CHAN_CONFIG_SINGLE)
        ?"single-ended":"differential",analogChannel,myAnalogRead(spiChannel,channelConfig,analogChannel-1));
}
else
{
    for(i=0; i<8; i++)
    {
        printf("MCP3008(CE%d,%s): analogChannel %d = %d\n",spiChannel,(channelConfig==CHAN_CONFIG_SINGLE)
            ?"single-ended":"differential",i+1,myAnalogRead(spiChannel,channelConfig,i));
    }
}
close (myFd) ;
return 0;
}

```

2- Código usado para leitura da temperatura:

```

#include<stdio.h>
#include<stdlib.h>
#define COMMAND "cat /sys/bus/w1/devices/28*/w1_slave > tmp.txt"
//#define COMMAND "cat file.txt > tmp.txt"

// le a temperatura
float temp_reader(){

    int c;
    int cp = 0;
    char number[10];
    int flag = 0;
    int i = 0;
    float temp;

    system(COMMAND);//lê e redireciona para um tmp
    FILE *fp;
    fp = fopen("tmp.txt","r");

    while((c = fgetc(fp)) != EOF){
        if (c =='\n') flag = 0; //evita fim de linha

        if (flag){
            number[i] = c;
            i++;
        }

        if ((cp=='t') && (c==' ')) flag = 1; // procura pelo sinal 't='

        cp = c;//armazena char anterior
    }
}

```

```
fclose(fp);
//system("rm tmp.txt") //removendo o tmp, caso deseje
temp = atoi(number)/1000.;
//printf("Text: %f ## \n", myvar/1000.);
return temp;
}

//a hard-written delay
void delay(){
unsigned int i;
for (i=0;i<100000000;i++) {}

}

//THE RUN
int main(){
    int i = 10;
    while(i--){
        printf("Temperatura: %.3f \n", temp_reader());
        delay();
    }
    return 0;
}
```