SISTEMA DE MONITORAMENTO VIA RFID

O “sistema de monitoramento via Rfid” tem como objetivo o monitoramento de equipamentos, veículos, animais, pessoas, etc, que utiliza das mais modernas tecnologias como rfid. RFID (*Radio Frequency Identification* – Identificação por Rádio Freqüência) é uma tecnologia utilizada para identificar, rastrear e gerenciar estes produtos sem contato e sem a necessidade de um campo visual. Composta por *transponders* (*RF tags*), leitores com antenas e microcontrolador ou outro tipo de controlador, RFID é uma tecnologia de identificação que utiliza a radiofreqüência para capturar os dados, permitindo que uma *tag* seja lida sem a necessidade de campo visual, através de barreiras e objetos tais como madeira, plástico, papel etc. Um sistema RFID digital funciona como um sistema poderoso de aquisição de dados em tempo real, com a vantagem de eliminação de intervenções humanas manuais e visuais, dinamizando assim o tempo de transições e assegurando eficiência e eficácia.

Com a ideia de implementar esta tecnologia no CBMSC, iniciou-se o projeto piloto de monitorar as viaturas no patio entrada/saída e o monitoramento dos equipamentos mais caros (gerador, ferramentas combinadas,etc) presente/ausente.

O projeto é desenvolvido em duas partes:

-Parte 1 (Monitorar viaturas);

-Parte 2 (Monitorar equipamentos);

Houve a necessidade de dividir o projeto em dois pela necessidade de cada equipamento que monitora. São equipamentos diferentes, porém a mesma tecnologia RFID.

Tecnologias utilizadas

Microcontrolador – Arduino Uno ;
- Shield Ethernet ;

Sensores/Tag – Reader ID-12LA/125kHz ;

Comunicação – Roteador Intelbras WRN 211;
- Modem 3g;

Projeto – Sistema de Monitoramento via Rfid

Resumo: Este projeto traz para o CBMSC a introdução a automação no cotidiano da corporação. Essa tecnologia aplicada tem inúmeras finalidades, para este projeto piloto iniciamos com o monitoramento das entradas/saídas das viaturas no quartel, hoje é feito manualmente o J9/J12 pela guarnição a central, com este equipamento de alta frequência será automatizada estas informações não necessitando da intervenção humana. Neste mesmo projeto piloto foi utilizada a tecnologia Rfid na parte interior do ABTR, porém com frequência bem menor que é utilizado na parte externa. O monitoramento dos equipamentos de dentro da viatura traz informações em tempo real se o equipamento está no local e se não está informa a hora exata que foi retirado da viatura. Um sistema de interface para o usuário foi desenvolvido com o tratamento destes dados.

1.Objetivo Geral: Implementar Monitoramento remoto via sensores Rfid de equipamentos e viaturas no CBMSC.

1.1Objetivos Específicos:
-Monitorar 24h equipamentos online de dentro da viatura;
-Monitorar J9/J12 das viaturas do quartel;
-Realizar levantamento de bens;

2.Funcionamento: O operador do COBOM, via interface irá ter a informação exata do J9/J12 da viatura sem que haja qualquer intervenção humana. Também terá o controle interno dos equipamentos da OBM, e podendo saber se foi retirado os equipamentos de dentro da viatura e por quanto tempo.

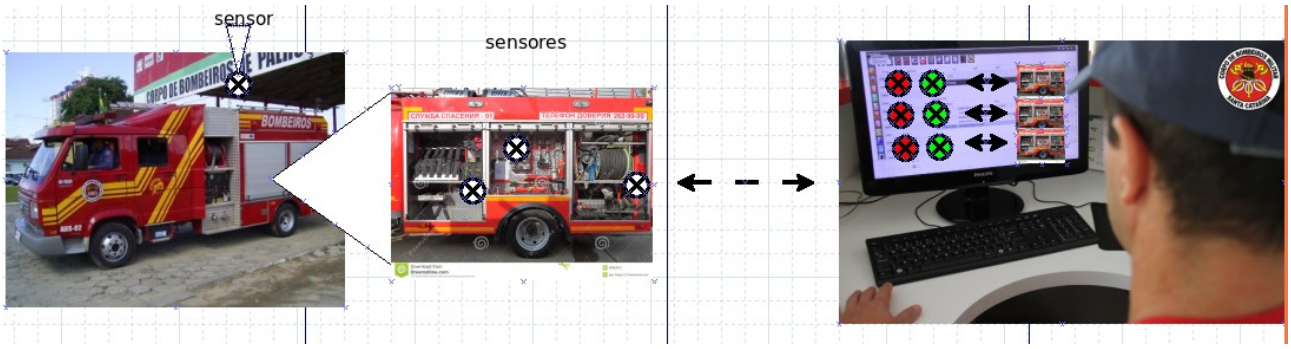
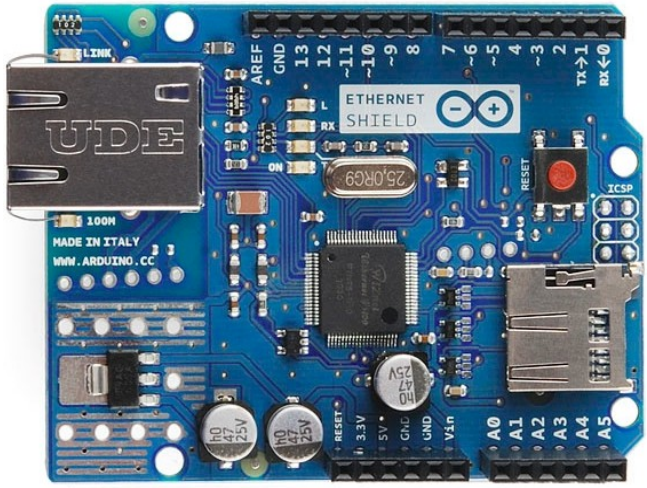


Fig 1. Diagrama Funcional do Projeto

2.1 Equipamentos:
2.1.1 Microcontrolador: Arduino Uno.

2.1.2.Shield Ethernet.



2.1.3 Sensores: ID-12LA/tag.

2.1.4 Roteador: Intelbras WRN 211/ Modem 3g.

3.Diagramas, Códigos e PCI.
3.1.Códigos: Na IDE do arduino foi desenvolvido um código para ler e imprimir nossas tags e comunicação xbee, fixado no código número de IP e protocolos de comunicação, segue anexo o código.

```
/*
  Software serial multple serial test

 Receives from the hardware serial, sends to software serial.
 Receives from software serial, sends to hardware serial.

 The circuit:
 * RX is digital pin 10 (connect to TX of other device)
 * TX is digital pin 11 (connect to RX of other device)

 Note:
 Not all pins on the Mega and Mega 2560 support change interrupts,
 so only the following can be used for RX:
 10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69

 Not all pins on the Leonardo support change interrupts,
 so only the following can be used for RX:
 8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).

 created back in the mists of time
 modified 25 May 2012
 by Tom Igoe
 based on Mikal Hart's example

 This example code is in the public domain.

 */
#include <SoftwareSerial.h>
#include <SPI.h>
#include <Ethernet.h>
#include <avr/pgmspace.h>

SoftwareSerial mySerial(2, 3); //RX, TX
/*****************************************/
byte mac[] = {
  0x90, 0xA2, 0xDA, 0x0E, 0x0C, 0xAC };;
IPAddress ip(10,193,191,246); //172.16.116.76.--IP IFSC
IPAddress subnet(255,255,255,0);
EthernetServer server(80);
/*****************************************/
unsigned char abtr66 =0;
void setup()
{
  pinMode(9, OUTPUT);
  digitalWrite(9, LOW);
  pinMode(6, OUTPUT);
  digitalWrite(6, LOW);
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  Serial.println("XBEE Example!");
   // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);
```

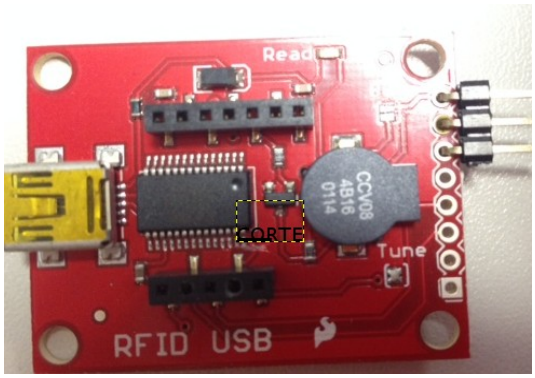
```
  /*****************************************/
  // Inicia a conexão Ethernet e o servidor:
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("Servidor iniciado em: ");
  Serial.println(Ethernet.localIP());
  /*****************************************/
}

void loop () {
  byte i = 0;
  byte val = 0;
  byte code[6];
  byte checksum = 0;
  byte bytesread = 0;
  byte tempbyte = 0;

  digitalWrite(9, HIGH);
  delay(1000);
  digitalWrite(9, LOW);

  if(Serial.available() > 0) {
    if((val = Serial.read()) == 2) {                // check for header
      bytesread = 0;
      while (bytesread < 12) {                      // read 10 digit code + 2 digit checksum
        if( Serial.available() > 0) {
          val = Serial.read();
          if((val == 0x0D)||(val == 0x0A)||(val == 0x03)||(val == 0x02)) { // if header or stop bytes
before the 10 digit reading
            break;                                  // stop reading
          }

          // Do Ascii/Hex conversion:
          if ((val >= '0') && (val <= '9')) {
            val = val - '0';
          } else if ((val >= 'A') && (val <= 'F')) {
            val = 10 + val - 'A';
          }

          // Every two hex-digits, add byte to code:
          if (bytesread & 1 == 1) {
            // make some space for this hex-digit by
            // shifting the previous hex-digit with 4 bits to the left:
            code[bytesread >> 1] = (val | (tempbyte << 4));

            if (bytesread >> 1 != 5) {              // If we're at the checksum byte,
              checksum ^= code[bytesread >> 1];     // Calculate the checksum... (XOR)
            };
          } else {
            tempbyte = val;                         // Store the first hex digit first...
          };
```

```
      bytesread++;                        // ready to read next digit
     }
    }

    // Output to Serial:

    if (bytesread == 12) {                // if 12 digit read is complete
     Serial.print("5-byte code: ");
     for (i=0; i<5; i++) {
      if (code[i] < 16) Serial.print("0");
      Serial.print(code[i], HEX);
      Serial.print(" ");
     }
     Serial.println();

     Serial.print("Checksum: ");
     Serial.print(code[5], HEX);
     Serial.println(code[5] == checksum ? " -- passed." : " -- error.");

     Serial.println();

    }
    bytesread = 0;
   }
   abtr66 = 1;
  }
  else
   abtr66 = 0;


 /*********************************************************************************/

EthernetClient client = server.available();
 if (client) {
  Serial.println("Novo Cliente");
  // Uma solicitação http termina com uma linha em branco
  boolean currentLineIsBlank = true;
  while (client.connected()) {
   if (client.available()) {
    char c = client.read();
    //Serial.write(c);
    // Se tiver chegado ao fim da linha (recebeu um novo
    // Caractere) e a linha estiver em branco, o pedido http terminou,
    // Para que você possa enviar uma resposta
    if (c == '\n' && currentLineIsBlank) {

     // Envia um cabeçalho de resposta HTTP padrão
     client.println("HTTP/1.1 200 OK");
     client.println("Content-Type: text/html");
     client.println("Connection: close");  // a conexão será fechada após a conclusão da resposta
     client.println("Refresh: 18");  // atualizar a página automaticamente a cada 5 segundos
     client.println();
```

```
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          /*************************************/
          client.println("<head>");
          client.println("<style>");
          client.println("table, th, td {");
          if(abtr66 == 0)
            client.println("background-color: #F73A3A;");
          else
            client.println("background-color: #45F25F;");
          client.println("border: 1px solid black;");
          client.println("border-collapse: collapse;}");

          client.println("th, td {");
          client.println("padding: 5px;");
          client.println("text-align: left;}");
          client.println("</style>");
          client.println("</head>");
          client.println("<body>");

          client.println("<table style=""width:10%"">");
          client.println("<tr>");
          client.println("<th>ABTR 66</th>");
          //client.println("<th>Last Name</th>");
          //client.println("<th>Points</th>");
          client.println("</tr>");
          client.println("</table>");
          client.println("</body>");
          client.println("</html>");
          /*************************************/

          //client.print("TAG 1: ");
          //for(i=0; i<5; i++){
            //client.print("<br> "+ code[i]);
//          client.print("<br>TAG 3: "+ tagID3);
//          client.println("</html>");
          //tagID = "";
          //tagID2 = "";
          //tagID3 = "";
          //}
          break;
        }
        if (c == '\n') {
          // Você está começando uma nova linha
          currentLineIsBlank = true;
        }
        else if (c != '\r') {
          // Você recebeu um caracter na linha atual.
          currentLineIsBlank = false;
        }
      }
    }
```

```
    // Dar tempo ao navegador para receber os dados
    delay(1);
    // Fecha a conexão:
    client.stop();
    Serial.println("Cliente desconectado");
  }
  /*****************************************************************************/
//  digitalWrite(6, HIGH);
//  delay(50);
//  digitalWrite(6, LOW);
}
```

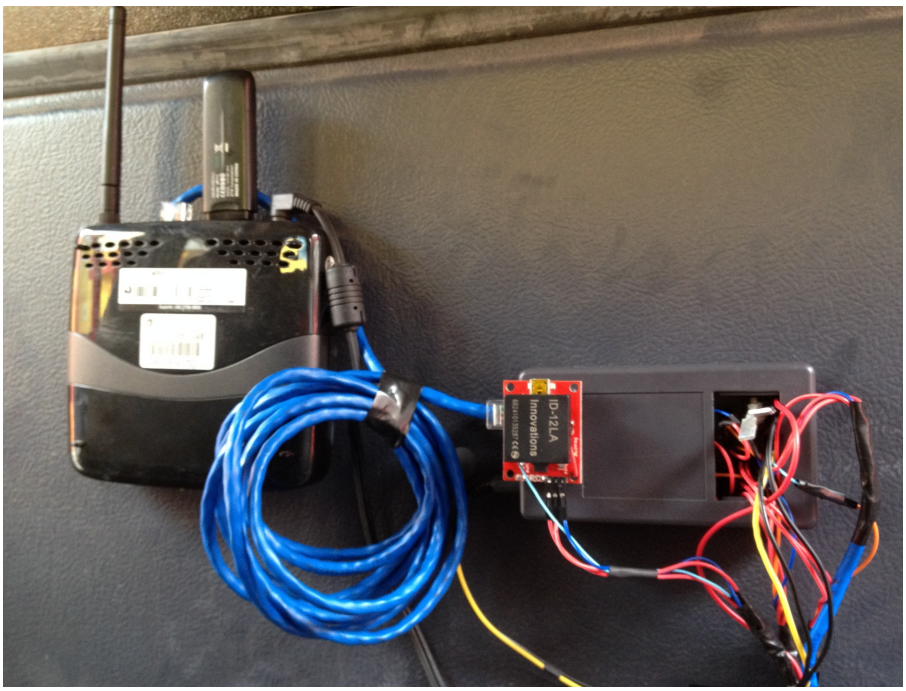
3.2.Diagramas: Foi feito diagramas elétricos para a conexão do sistema especificamente para o Caminhão da marca AVECO. Lembrando que cada viatura terá um diagrama e suas especifidade.
3.3.PCI: A princípio foi desenvolvida, soldada e testada uma pci pela nossa euipe para regular a tensão e as conexões dos sensores. Obtivemos sucesso com o circuito que também seguirá a seguir uma cópia em anexo. Para o uso dos sensores de rfid tivemos que romper uma trilha para forçar ele resetar a placa e ler as nossas tags. Segue a foto da trilha cortada :



[TUTORIAL - 1° Parte: Gravação do Arduino e montagem do sistema.]

Esta etapa contem a gravação do código no arduino, teste com os sensores, ligar os fios em seus pinos e por último a montagem na viatura e a fixação das tags nos equipamentos a serem monitorados.

Foi montado um projeto piloto em uma viatura do CBMSC, monitorando apenas para teste, o controle de aberturas das portas da viatura. Segue, algumas fotos do equipamento instalado:



O sistema de tratamento dos dados deverá será desenvolvido ainda, realizando uma interface para os usuários.

[2° Parte: Montagem do Sistema ABTR66.]

Após o desenvolvimento e a montagem do sistema de monitoramento Rfid na viatura AR66, como projeto piloto, e obtido um bom resultado. Começou-se a montagem do sistema na viatura ABTR66.

Os equipamentos que foram instalados na viatura: arduino uno, shield ethernet, sensores rfid, shield regulador de tensão, roteador wifi.

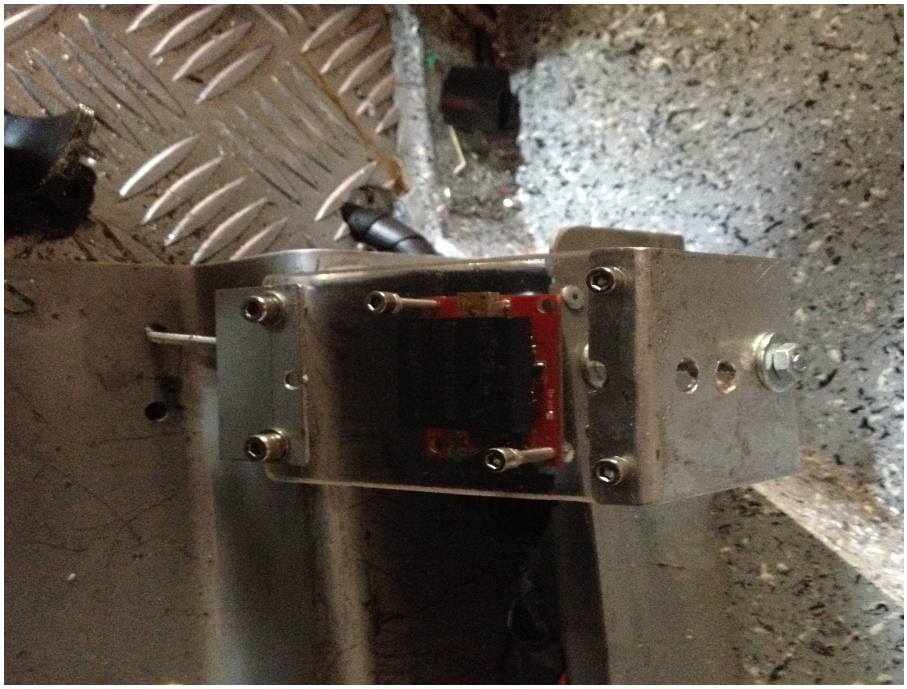
Muitas adaptações foram realizadas na viatura, devido ao alto índice de interferência no

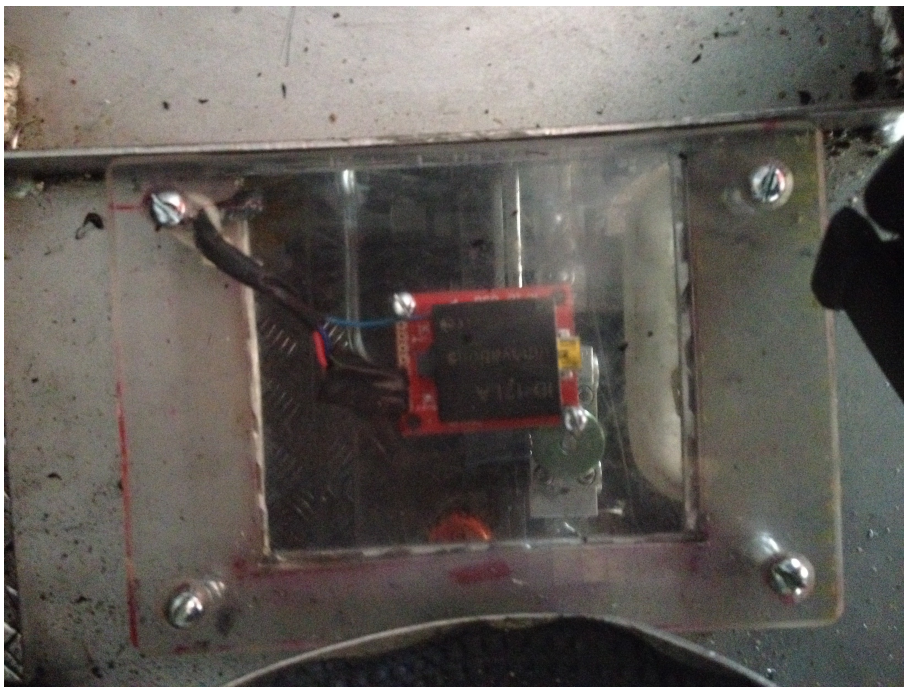
metal na tecnologia rfid, e ao cotidiano da guarnição.

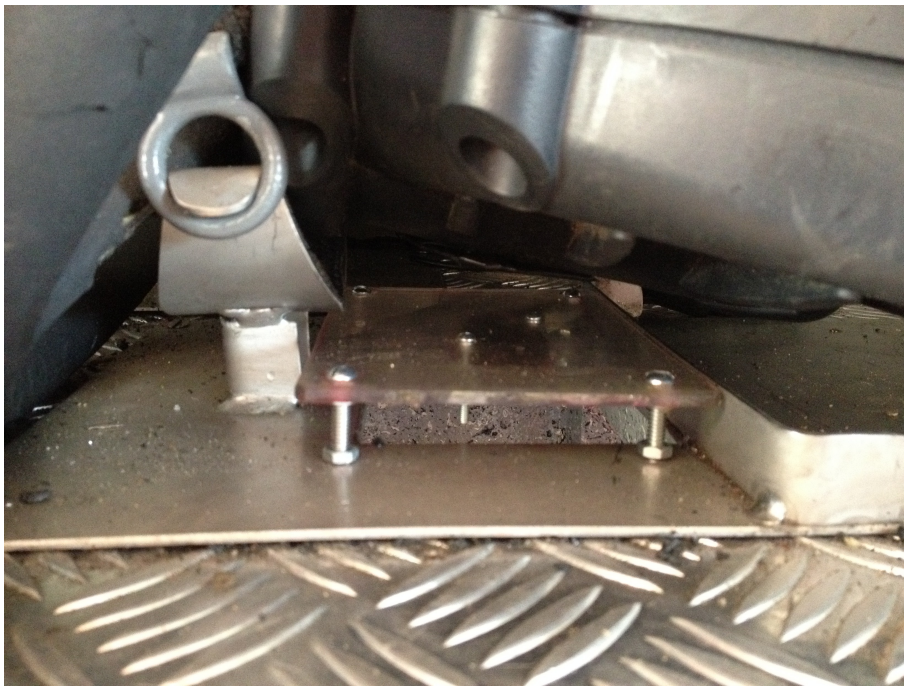
O sistema esta sendo alimentado por duas baterias de 12V em série e paralelo com mais duas baterias da mesma voltagem que são próprias do caminhão. Como nosso Sistema necessita de uma voltagem menor utilizamos placa de regulador de tensão para o controle dessa voltagem.

Foi desenvolvido e usinado um suporte específico para essa fixação das novas baterias. Imagens a seguir;

Foi desenvolvido peças de metal, policarbonato e EVA como aparatos para conseguir realizar estas adaptações. A seguir, imagens sobre estas adaptações:







[3° Parte: Algumas Conclusões]

O sistema está lendo e enviando para a web como o esperado, com um time de aproximadamente 10s. Relatórios estão sendo enviados para o e-mail pessoal, quando uma das ferramentas foram retiradas e quando o sensor J9/J12 realiza a leitura. Futuramente, será criado um banco de dados com todos os códigos referente as ferramentas e as viaturas.

A seguir imagem da interface com o usuário:

Nesta imagem, apresenta as ferramentas que estão presentes na viatura e mostra que a viatura está no pátio.

O sensor de nível de água esta em desenvolvimento, logo estará disponível a sua leitura.

- **Problemas Encontrados**

1° - Grande interferência pelo metal na tecnologia aplicada;
2° - Problemas com acesso ao no-ip;
3° - Controle de posição das ferramentas e viaturas e;
4° - Costume e cultura da GU.