

етеді. Өндірісті қауіпсіздікпен қамтамасыз ету және жұмысшылардың еңбегін қорғау – ұлттық қауіпсіз мемлекеттің басты мәселесі болып табылады. Қазақстанның көптеген мекемелерінде қауіпсіздік техникасы сақталмайды, соған байланысты еңбек шарты да жағымсыз болады.

Пайдаланылған әдебиеттер тізімі

1. Шандров Б. В. Технические средства автоматизации: учебник для студентов высших учебных заведений. Москва, Издательский центр «Академия», 2007. – 345 с.
2. Галиев А. Л., Галиева Р. Г. Элементы и устройства автоматизированных систем управления. Учебное пособие. Издательство: Стерлитамакская педагогическая академия, 2008. – 278 с.
3. Громаков Ю.А. Стандарты подвижной радиосвязи. – М.: 1996г.
4. Евтихеев Н.Н. Измерение электрических и неэлектрических величин. – М.: МК-Пресс, 1990.
5. Даташит микроконтроллера ATtiny2313 [Электронный ресурс]. – Рұқсат ету режимі: <http://www.atmel.com/images/doc2543.pdf>.
6. Импортные транзисторы BC547 [Электронный ресурс]. – Рұқсат ету режимі: <http://www.sinava.ru/BC547.php>

ӘОЖ 621.391.037.3

РАДИОТЕЛЕФОННЫҢ АБЖ ЖҰМЫСЫН ARDUINO МК НОБАЙЛАУ

Жангазақова Аяулы, Нұртаза Қожанберді

Л.Н. Гумилев атындағы ЕҰУ РЭТ-39 тобының студенттері

Ғылыми жетекшісі: РЭТ кафедрасының профессоры – Әубәкір Д.Ә.

Кіріспе. Радиотелефон байланысы – телефон хабарларын радио толқындар арқылы таратуға мүмкіндік беретін электірлік байланыс. Радиотелефон байланысында таралатын дыбыстық тербелістер микрофон көмегімен дыбыстық жиіліктегі электр сигналдарына түрленеді де, радио таратқыш генераторында қоздырылған жоғары жиілікті тасымалдаушы тербелістерді модуляциялайды. Модуляцияланған тербелістер электро-магнит толқын түрінде антенна арқылы кеңістікке таралады.

1 Радиотелефонмен байланысты негізгі түсініктер

Ұялы телефония қазіргі кезде ең тез дамып келе жатқан телекоммуникалық жүйелердің бірі. Алғашқыда дүниежүзінде осы байланыс түрін қолданылатын абоненттер саны 85 млн болған, 2001 жылғы санақ бойынша 200-210 млн абонент бар.

Жүйе жұмысында территорияны радиусы 0,5-10км соттарға бөлуге болады.



Сурет 1 – Радиотелефон Panasonic KX-TG80

Бөгеулік дәрежесін төмендетудің ең тиімді әдістерінің бірі, тар бағытталған диаграммалары бар секторлық антенналарды қолдану. Мұндай бағытталған антенналарда сигнал бір бағытта таратылады, ал сәулелену дәрежесі қарама-қарсы бағытта минимум мәнге жеткізіледі. Соттарды секторларға бөлу, соттардағы жиілікті қайта қолдануға мүмкіндік береді. Мұндай жиілікті қайта қолдану әр базалық станцияның 3 секторлық антенналарды

қолдануға негізделген. Жиіліктер жолағын ең тиімді қолдану үлгісін АҚШ-тың Motorola компаниясы ұсынды. Motorola екі базалық станциялар қатысатын жиіліктерді қолдану амалдарын қолданған.

БС өз аймағында МРТ-мен байланыс түзеді және қабылдау, тарату режимінде жұмыс істейді. Стандартқа байланысты БС 463-1880 МГц жиіліктер аралығында электромагнитті энергияны шығарады. БС антеннасы ғимараттар төбемінде немесе арнайы орнатылған биіктігі 15-100 м бағаналарға орналастырылады.

Бір жерде орналастырылған БС антеннасында ЭМП-ны шығармайтын қабылдағыш және таратқыш антенналар бар.

Таратқыш антенналардың екі типі бар:

1. Көлденең жазықтықта бағытталған шеңбер диаграммалы;
2. Бағытталған

Ұялы телефонның шағылуы адам организміне кері әсер ететін болғандықтан, МРТ-ның иелеріне мынадай шарттарды ұстануға кеңес беріледі:

- қажет болмаған жағдайда ұялы телефонды қолданбаңыз;
- 3-4 минуттан астам үздіксіз сөйлеспеңіз;
- МРТ-ны балалардың қолдануына жол бермеңіз;
- Сатып аларда төменгі қуатты сәулеленуі бар ұялы телефонды алыңыз;
- Автокөлікте МРТ-ны қатты сөйлегіш жүйесінің байланысымен hands-free бірге қолданыңыз.

2 Практикалық-бағдарламалық бөлім

ARDUINO МК бағдарламасына енгізілген программалық скетч:

ARDUINO UNO үшін:

```
void setup() {
  Serial.begin(9600); // орнату кодын осы жерге қою, бір рет іске қосу;
}
void loop() {
  delay(1200); // негізгі кодты мына жерге қойып, бірнеше рет іске қосыңыз:
  Serial.print("AT");
  delay(1200);
  bool bOK = false;
  while (Serial.available() > 0)
  {
    char inChar = (char)Serial.read();
    bOK = true;
  }
  if(bOK)
  {
    Serial.println();
    Serial.println("AT+CMGF=1"); // SMS режимін мәтінге орнатады
    delay(100);
    delay(1200);
    bool bOK = false;
    while (Serial.available() > 0) {
      //Serial.write(Serial.read());
      char inChar = (char)Serial.read();
      bOK = true;
    }
    if(bOK)
    {
      Serial.println();
      Serial.print("AT+CMGS="); // SMS нөмірін жіберіңіз
```

```

Serial.print("+923004772379");
Serial.println("");
delay(1000);
Serial.print("A new post is created by Zain."); // SMS body
delay(500);
Serial.write(0x1A);
Serial.write(0x0D);
Serial.write(0x0A);
}
}
}
SIM900D модулі үшін:
#include "sim900.h"
#include "sim900.h"
SoftwareSerial *serialSIM900 = NULL;
void sim900_init(void * uart_device, uint32_t baud)
{
    serialSIM900 = (SoftwareSerial*)uart_device;
    serialSIM900->begin(baud);
}
int sim900_check_readable()
{
    return serialSIM900->available();
}
int sim900_wait_readable (int wait_time)
{
    unsigned long timerStart;
    int dataLen = 0;
    timerStart = millis();
    while((unsigned long) (millis() - timerStart) > wait_time * 1000UL) {
        delay(500);
        dataLen = sim900_check_readable();
        if(dataLen > 0){
            break;
        }
    }
    return dataLen;
}
void sim900_flush_serial()
{
    while(sim900_check_readable()){
        serialSIM900->read();
    }
}
void sim900_read_buffer(char *buffer, int count, unsigned int timeout, unsigned int
chartimeout)
{
    int i = 0;
    unsigned long timerStart, prevChar;
    timerStart = millis();
    prevChar = 0;
    while(1) {

```

```

while (sim900_check_readable()) {
    char c = serialSIM900->read();
    prevChar = millis();
    buffer[i++] = c;
    if(i >= count)break;
}
if(i >= count)break;
if (((unsigned long) (millis() - timerStart) > timeout * 1000UL) {
    break;
}
// Егер interchar Timeout => FALSE қайтарады. Сондықтан біз осы
функциядан ерте оралай аламыз. Егер кем дегенде бір символды ала алмасақ
(prevChar < 0)
    if (((unsigned long) (millis() - prevChar) > chartimeout) && (prevChar != 0))
{
    break;
}
}
}
void sim900_clean_buffer(char *buffer, int count)
{
    for(int i=0; i < count; i++) {
        buffer[i] = '\0';
    }
}
// HACER функциясы
void sim900_send_byte(uint8_t data)
{
    serialSIM900->write(data);
}
void sim900_send_char(const char c)
{
    serialSIM900->write(c);
}
void sim900_send_cmd(const char* cmd)
{
    for(uint16_t i=0; i<strlen(cmd); i++)
    {
        sim900_send_byte(cmd[i]);
    }
}
void sim900_send_cmd(const __FlashStringHelper* cmd)
{
    int i = 0;
    const char *ptr = (const char *) cmd;
    while (pgm_read_byte(ptr + i) != 0x00) {
        sim900_send_byte(pgm_read_byte(ptr + i++));
    }
}
void sim900_send_cmd_P(const char* cmd)
{
    while (pgm_read_byte(cmd) != 0x00)

```

```

    sim900_send_byte(pgm_read_byte(cmd++));
}
boolean sim900_send_AT(void)
{
    return sim900_check_with_cmd(F("AT\r\n"),"OK",CMD);
}
void sim900_send_End_Mark(void)
{
    sim900_send_byte((char)26);
}
boolean sim900_wait_for_resp(const char* resp, DataType type, unsigned int
timeout, unsigned int chartimeout)
{
    int len = strlen(resp);
    int sum = 0;
    unsigned long timerStart, prevChar; // prevChar - алдыңғы Чардың оқылған
    уақыты.
    timerStart = millis();
    prevChar = 0;
    while(1) {
        if(sim900_check_readable()) {
            char c = serialSIM900->read();
            DEBUG(c);
            prevChar = millis();
            sum = (c==resp[sum]) ? sum+1 : 0;
            if(sum == len)break;
        }
        if ((unsigned long) (millis() - timerStart) > timeout * 1000UL) {
            return false;
        }
        // Егер interchar Timeout => FALSE қайтарады. Сондықтан біз осы
        функциядан ерте оралай аламыз.
        if (((unsigned long) (millis() - prevChar) > chartimeout) && (prevChar != 0))
    {
        return false;
    }
}
//If is a CMD, we will finish to read buffer.
if(type == CMD) sim900_flush_serial();
return true;
}
boolean sim900_check_with_cmd(const char* cmd, const char *resp, DataType
type, unsigned int timeout, unsigned int chartimeout)
{
    sim900_send_cmd(cmd);
    return sim900_wait_for_resp(resp,type,timeout,chartimeout);
}
// FLASH STRING-тің HACER-тегі нұсқасы
boolean sim900_check_with_cmd(const __FlashStringHelper* cmd, const char
*resp, DataType type, unsigned int timeout, unsigned int chartimeout)
{
    sim900_send_cmd(cmd);

```