

№15. AMALIY MASHG'ULOT

AQLLI SHAHAR BOSHQARUVCHISINI BOSHQARISH UCHUN SHAXSIY KOMPYUTERDA BAJARILADIGAN DASTURNI ISHLAB CHIQISH.

Ishning maqsadi: Arduino asosidagi shaxsiy kompyuter orqali aqlli shaharni boshqarish tizimlarining apparat va dasturiy ta'minotini o'rganish. Amaldagi tizimning tarkibiy qismlari haqida texnik xususiyatlar va umumiy ma'lumotlarni o'rganish.

Amaliy mashg'ulot № 15.1. Arduino-ni kompyuterdan boshqarish.

Mashg'ulotning maqsadi: USB orqali kompyuterdan Arduino-ni qanday boshqarish mumkin. **Serial.available()** va **Serial.read()** funksiyalaridan foydalangan holda klaviaturadan Arduino-da LEDni qanday yoqish va servo motorni boshqarishni sizga aytib beramiz . Bu funksiyalar ketma-ket port orqali mikrokontrollerga biron-bir buyruq kelishini tekshiradi va keyin kompyuterdan Serial monitorga kiritilgan kiruvchi buyruqlarni o'qiydi.

Umumiy ish rejasi:

1. Serial monitordan Arduino boshqaruvi
2. Arduino-ni klaviatura orqali qanday boshqarish mumkin
3. USB orqali Arduino LED boshqaruv kodi
4. Kompyuterdan Arduino servo boshqaruv kodi



15.1-rasm. Arduino nazorati kompyuter orqali

Serial.available () funksiyasi ketma-ket portdan o'qish uchun mavjud baytlar sonini oladi. Bular kompyuterdan yuborilgan va ketma-ket port buferiga yoziladigan baytlardir. Arduino Serial monitor buferi maksimal 64 baytni saqlashi mumkin. Funktsiya Bluetooth moduli Arduino bilan o'zaro aloqada bo'lganda ham qo'llaniladi va dizayn bosqichida qurilmani disk raskadrovka qilishda foydalidir.

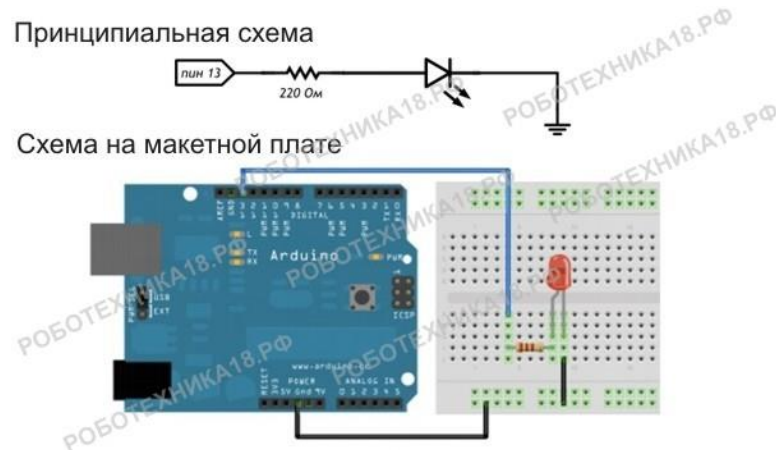
Arduino qayig'i kabi Bluetooth bilan boshqariladigan turli xil qurilmalarni sinab ko'rish va sozlashda kompyuter orqali LED va servo motorni qanday boshqarishni bilish foydali bo'ladi. Shuning uchun, endi biz USB kabeli orqali kompyuter orqali

servoni oddiy boshqarishni ko'rib chiqamiz. Shu bilan birga, monitor orqali nafaqat raqamlar, balki lotin alifbosidagi harflar ham yuborilishi mumkin.

Arduino-ni klaviaturadan qanday boshqarish mumkin

Ushbu mashg'ulot uchun sizga kerak bo'ladi:

- Arduino Uno / Arduino Nano / Arduino Mega;
- plata;
- LED va 220 Om qarshilik;
- servo;
- "ota-ota" va "ota-ona" simlari.



15.2-rasm. Kompyuterdan Arduino UNO da LEDni boshqarish

Arduino UNO uchun qolgan aktuatorlarni o'zingizning xohishingizga ko'ra olishingiz va USB kabelidan foydalangan holda kompyuter orqali o'zingiz boshqarishga harakat qilishingiz mumkin. Birinchi mashg'ulotdan boshlab paneldagi bitta LED bilan sxemani Arduino nima va undan qanday foydalanishni quring va LED bilan sxemani yig'gandan so'ng, mikrokontrollerga quyidagi eskizni yuklang.

Eskiz. Arduino va kompyuter o'rtasidagi o'zaro ta'sir

```
int val; // o'zgaruvchi uchun kontrollerdagi bo'sh xotira

void setup() {
  Serial.begin(9600); // ketma-ket portni ulash
  pinMode (13, OUTPUT ); // 13-pinni chiqish sifatida e'lon qilish
}

void loop() {
  // biron-bir buyruq kelishini tekshiring
  if ( Seriya . mavjud () ) {
```

```

val = Serial.read (); // val o'zgaruvchisi qabul qilingan buyruqqa teng

if (val == '1') { digitalWrite (13, HIGH );} // LEDni 1 marta yoqing
if (val == '0') { digitalWrite (13, LOW );} // 0 da LEDni o'chiring

}

}

```

Kod uchun tushuntirishlar:

1. Serial.available() funksiyasi kompyuterdan buyruqlarni tekshiradi;
2. Serial.read() funksiyasi Serial monitor tomonidan qabul qilingan ma'lumotlarni o'qiydi.
3. if (val == '1') shartli bayonotida "moslik" qo'sh tenglik belgisi mavjud va qavs ichida bitta tirnoq ishlatilishi kerak.



15.3-rasm. Arduino UNO-da kompyuterdan servo boshqaruv

Sxemani yig'gandan so'ng, quyidagi eskizni taxtaga yuklang
Eskiz. Kompyuterdan servo boshqaruv

```

#include < Servo .h> // servo uchun kutubxonani qo'shing
Servo servo // "servo" tipidagi servo o'zgaruvchini e'lon qilish
int val; // o'zgaruvchi uchun kontrollerdagi bo'sh xotira

void setup() {

    Serial.begin(9600); // ketma-ket portni ulash

    servo.attach(11); // servoni 11-pinga ulang
}

```

```

void loop() {
  // biron-bir buyruq kelishini tekshiring
  if (Serial.available()) {

    val = Serial.read(); // val o'zgaruvchisi qabul qilingan buyruqqa teng

    if (val == '1') { servo.write(10); } // 1da servoni 10 ga aylantiring
    if (val == '2') { servo.write(45); } // 2 da servoni 45 ga aylantiring
    if (val == '3') { servo.write(90); } // 3 da servoni 90 ga aylantiring

  }
}

```

Kod uchun tushuntirishlar:

1. agar (Serial.available()>0) seriyali monitorda ma'lumotlarning ko'rinishini tekshirishning bunday variantini topishingiz mumkin .
2. buyruqlar sifatida siz nafaqat raqamlarni, balki harflarni ham ishlatishingiz mumkin. Harf katta yoki kichik bo'lishi muhim emas.

Dastur yozish algoritmi:

1. Muammoni oydinlashtirish;
2. Nazariy manbalar va foydali materiallarni izlash;
3. Zarur bo'lganda kutubxonani ulash;
4. Dastur kodini yozish;
5. Kompilyatsiya.

Hisobotda quyidagilar bo'lishi kerak:

1. Amaliyot nomlari va ishning maqsadi.
2. Breadboard va tadqiqot tushunchasi.
3. Tadqiqotda foydalaniladigan texnik vositalarning xarakteristikalar.
4. Dastur kodini va kompilyatsiya skrinshotlarini yozish.
5. Bajirilgan vazifa bo'yicha foto / video hisobot.
6. Xavfsizlik savollariga javob yozing (qo'lda).

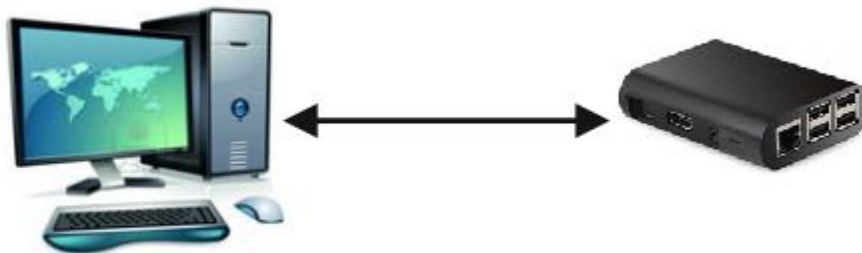
№15.2. USB orqali kompyuter va Arduino o'rtasida ketma-ket aloqa.

Mashg'ulotning maqsadi: Mashg'ulotda men ModBus RTU protokoli bo'lgan tarmoqlarda Arduino slave kontroller uchun dasturiy ta'minotni ishlab chiqish uchun kutubxonani taqdim etaman. Misol tariqasida, men UART interfeysi orqali boshqaruvchining kompyuterga ulanishini amalga oshiraman.

Umumiy ish rejasi:

1. Qurilmaning sxematik sxemasini tuzing.
2. Ushbu sxemani lehimsiz taxtada amalga oshiring.

3. Mikrokontroller uchun dastur yozing.
4. Qurilmani sinab ko'ring.



15.4-rasm. USB orqali kompyuter va Arduino o'rtasida ketma-ket aloqa

Keling, ushbu protokol yordamida kontrollerimizni kompyuterga ulaymiz. Buning uchun men oddiy kutubxona ishlab chiqdim. Avvalo, men uning funktsiyalari va imkoniyatlarini batafsil tasvirlab beraman.

Tiny_ModBusRTU_Slave kutubxonasi.

Tiny_ModBusRTU_Slave kutubxonasini ushbu havoladan yuklab olish mumkin:

[Ro'yxatdan o'ting](#) va to'lang. Faqat 60 rubl. saytning barcha resurslariga kirish uchun oyiga!

Kutubxona sizga ModBus RTU protokoli yordamida qul boshqaruvchi dasturini osongina amalga oshirish imkonini beradi. UART, RS-485, RS-422, RS-232 va boshqalar interfeyslari bilan birgalikda foydalanish mumkin. RS-485 kabi uchinchi davlat avtobus interfeyslarida transmitter boshqaruvini qo'llab-quvvatlaydi.

Kutubxona nomidagi "Kichik" (kichik) so'zi minimal funktsiyalar to'plami amalga oshirilganligini ta'kidlaydi. Xolding registrlari bilan ishlash uchun faqat 3 ta funktsiya kodlari. Biroq, ko'pchilik kontrollerlar aynan shu operatsiyalar to'plamini qo'llab-quvvatlaydi.

Kutubxona o'z vazifalarini parallel jarayonda bajaradi. Tarmoq orqali barcha aloqalar taymer uzilishida sodir bo'ladi. Asosiy tsiklda dastur oddiygina ModBus registrlar jadvalidan foydalanadi.

Tiny_ModBusRTU_Slave kutubxonasi konstruktor va faqat bitta funktsiyaga ega.

```
sinf Tiny_ModBusRTU_Slave {
```

```
public:
```

```
Tiny_ModBusRTU_Slave(byte adress, byte timeOut, unsigned int *  
holdingRegTable, unsigned int lengthTable); // konstruktor
```

```
Tiny_ModBusRTU_Slave(byte adress, byte timeOut, unsigned int *  
holdingRegTable, unsigned int lengthTable, byte directPin); // konstruktor
```

```
void update(); // ma'lumotlarni yuklash
```

```
boolean flagRead; // belgi ma'lumotlari o'qildi
```

```
boolean flagWrite; // belgi ma'lumotlari yozildi
```

```
};
```

Konstruktor .

```
Tiny_ModBusRTU_Slave(bayt manzili, bayt vaqti tugashi, unsigned int *  
holdingRegTable, unsigned int lengthTable, bayt directPin)
```

Quyidagi parametrlarga ega ob'ekt yaratadi:

- address - tarmoqdagi tobe qurilmaning manzili. 1 dan 247 gacha qiymatga ega bo'lishi mumkin.
- timeOut - javob yuborishdan oldin vaqtni pauza qilish (jim qilish). Qiymat kamida 3,5 baytni uzatish uchun zarur bo'lgan vaqt bo'lishi kerak. Bu timeOut * update() funksiyasiga qo'ng'iroq qilish davri sifatida hisoblanadi.
- holdingRegTable – massivga ko'rsatgich (massiv nomi) – saqlash registrlari jadvallari.
- longTable - saqlash registrlari jadvalining o'lchami.
- DirectPin - to'g'ridan-to'g'ri qurilmaning transmitter ishlashini yoqishning pin raqami. Transmitter uchta holatga ega bo'lgan avtobus interfeyslari uchun ishlatiladi. Masalan, RS-485. Parametr ixtiyoriy. Agar mavjud bo'lmasa, chiqish ishlatilmaydi.

```
Tiny_ModBusRTU_Slave slave(1, 12, regTable, 12); // ob'ekt yaratish, manzil 1,  
kutish vaqti 6 ms, regTable massivi, o'lcham 12
```

Usul.

update() - ma'lumotlarni yuklash. Yagona sinf funktsiyasi. Bu erda birjaning barcha boshqaruvi amalga oshiriladi. Funktsiya parallel jarayonda, masalan, taymer uzilishida muntazam ravishda chaqirilishi kerak.

```
// uzilishni qayta ishlovchi 500 mks
```

```
void timerInterrupt() {  
    slave.update();
```

```
}
```

update() usuli tarmoq aloqasini to'liq nazorat qiladi. Asosiy tsikl dasturi faqat regTable massivi (saqlash registrleri jadvali) bilan ishlaydi. Massiv ma'lumotlari o'qish va yozish uchun tarmoq ustasiga mavjud.

Har holda, men asosiy qurilmaning saqlash registrlariga kirishini ko'rsatuvchi 2 ta belgi qo'shdim.

- flagRead - ma'lumotlarni o'qish belgisi. Faol holat (to'g'ri) asosiy qurilma kamida bitta ushlab turish registrini o'qiganligini ko'rsatadi. Qayta ishlash jarayonida atribut asosiy dasturda qayta o'rnatilishi kerak.
- flagWrite - ma'lumotlarni yozish uchun bayroq. Faol holat kamida bitta registr yozilganligini bildiradi. Bular. jadval ma'lumotlari o'zgartirildi. Atribut asosiy dasturda qayta o'rnatilishi kerak.

Kutubxona funktsiya kodlarini qo'llab-quvvatlaydi.

Funktsiya kodi	Ism	Tavsif
03	READ HOLDING REGISTERS	Bir yoki bir nechta ash registrlarining hatlarini o'qish
06	FORCE SINGLE REGISTER	Bir xolding reestriga ng
16	FORCE MULTIPLE REGISTERS	Bir nechta ushlab vchi registrarni ketma- yozish

Quyidagi xato kodlari qayta ishlanadi.

Xato kodi	Ism	Tavsif
01	ILLEGAL FUNCTION	Funktsiya kodi rollerda qo'llab- vatlanmaydi
02	ILLEGAL DATA ADDRESS	Ma'lumotlar manzili 'g'ri

Kutubxona translyatsiya almashinuvini qo'llab-quvvatlaydi, bunda asosiy qurilma 0 ga teng manzilni yaratadi.

Har bir kadr uchun maksimal bayt soni 64 bayt.

Tiny_ModBusRTU_Slave kutubxonasidan foydalanish.

Arduino slave kontroller dasturini amaliy amalga oshirish uchun sizga quyidagilar kerak:

- Tiny_ModBusRTU_Slave kutubxonasini ulang;
- massiv yaratish - jadvalni ro'yxatga olish;
- Tiny_ModBusRTU_Slave ob'ektini yaratish;
- UART parametrlarini o'rnatish (Serial sinf);
- taymer uzilishini amalga oshiring va uning ishlov beruvchisiga update() funksiyasini kiriting.

```
#include <Tiny_ModBusRTU_Slave.h>
#include <TimerOne.h>
unsigned int regTable[10];
Tiny_ModBusRTU_Slave slave(1, 8, regTable, 10); // yaratish ob'ekt ,
manzil 1, kutish vaqti 4 ms , regjadval massivi , o'lcham 10
void setup() {
    Timer1.initialize(500); // ishga tushirish taymer 1, davr 500 mks
```

```
    Timer1.attachInterrupt(timerInterrupt, 500); //o'rnatish ishlov beruvchi xalaqit beradi
```

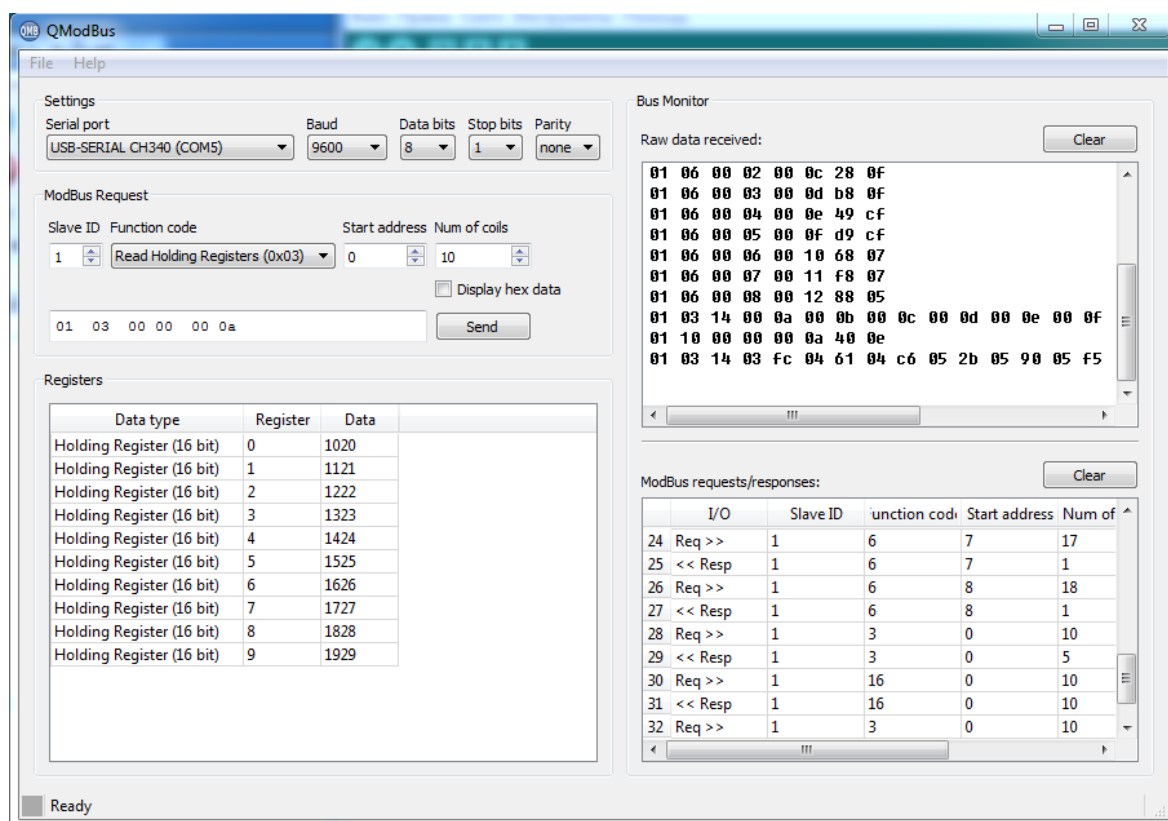
```
    Serial.begin(9600);
}
void loop() {
}
```

```
// ishlov beruvchi xalaqit beradi
```

```
void timerInterrupt() {
    slave.update();
}
```

loop() bu eskiz halqa bo'sh qoladi. Biroq, dastur ishlaydi. Agar siz platani ModBus tarmog'iga ulasangiz, u holda master kontroller regTable massividan ma'lumotlarni yozish va o'qish imkoniyatiga ega bo'ladi. Yana bir narsa shundaki, boshqaruvchi boshqa hech narsa qilmaydi. 10 registr uchun tarmoq saqlash qurilmasi chiqdi.

Keling, ushbu oddiy eskizni yuklaymiz va qul boshqaruvchisi qanday ishlashini ko'rib chiqamiz.



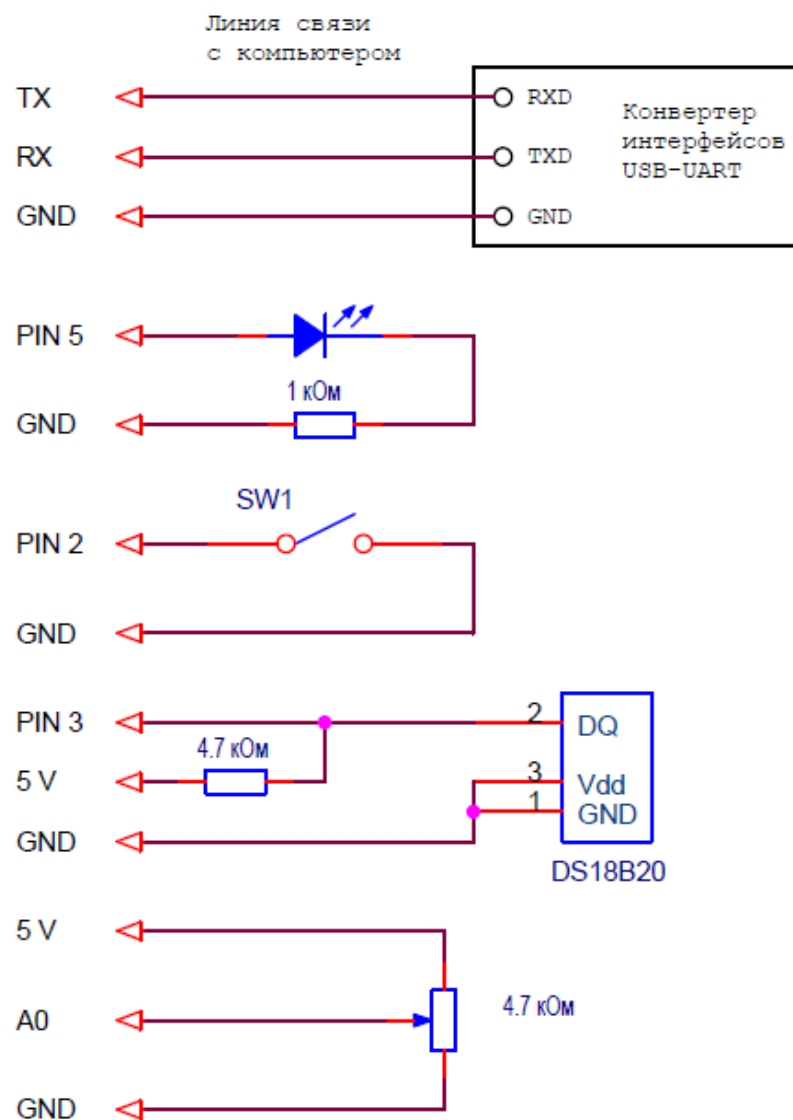
15.5-rasm. Kompyuter va Arduino uchun QModBus kompyuter interfeysi .

Sinov uchun men ModBus master emulyatoridan foydalandim. Internetda bunday dasturlar juda ko'p. Lekin o'zimga kerakli narsani katta qiyinchilik bilan topdim.

Ba'zi ModBus emulyatorlari pullik, boshqalari murakkab, qo'shimcha dasturlarni o'rnatishni talab qiladi. Ba'zilar faqat bitta registrning ishlashiga ruxsat beradi. Men o'zim uchun ishlaydigan emulyatorga duch keldim. Keyin ma'lum bo'ldiki, har bir jo'natishda u USB-USRT konvertorining DTR signalida impuls hosil qiladi va Arduino platasini qayta o'rnatadi. Men buni darhol tushunmadim.

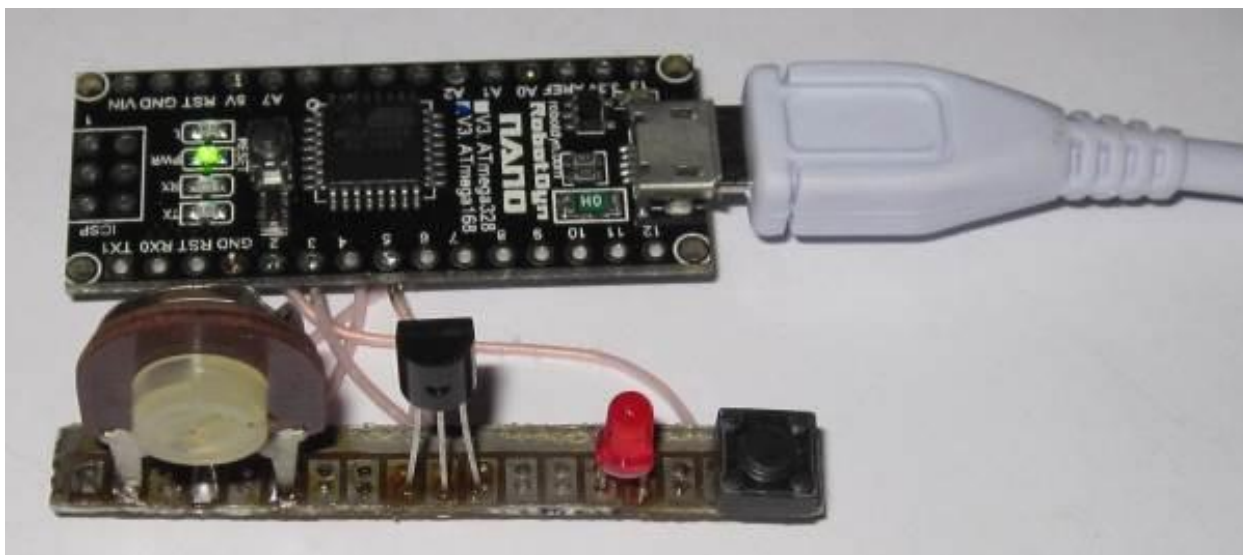
Oxir-oqibat, men uchun ishlaydigan ModBus RTU master emulyatorini topdim. Bu QModBus emulyatori. Men buni tasvirlamayman. Dastur oddiy va intuitivdir.

QModBus emulyatori "GNU General Public License GPL" ostida nashr etiladi, bu esa dasturni erkin qayta tarqatish imkonini beradi. Shuning uchun, men dasturni saytga joylashtiraman: [QModBus-0.2.1-win32](#).



15.6-rasm. Pinout DS 18 B 20 va USB - UART

QModBus-dan foydalanib, men kutubxonaning barcha funktsiya kodlarini tekshirdim, turli ma'lumotlarni yozdim va o'qidim va xato kodlari to'g'ri ishlanganligiga ishonch hosil qildim.



15.7-rasm. ModBus RTU mahalliy boshqaruvchisini joriy qilish

Albatta, dasturni o'zgartirish kerak.

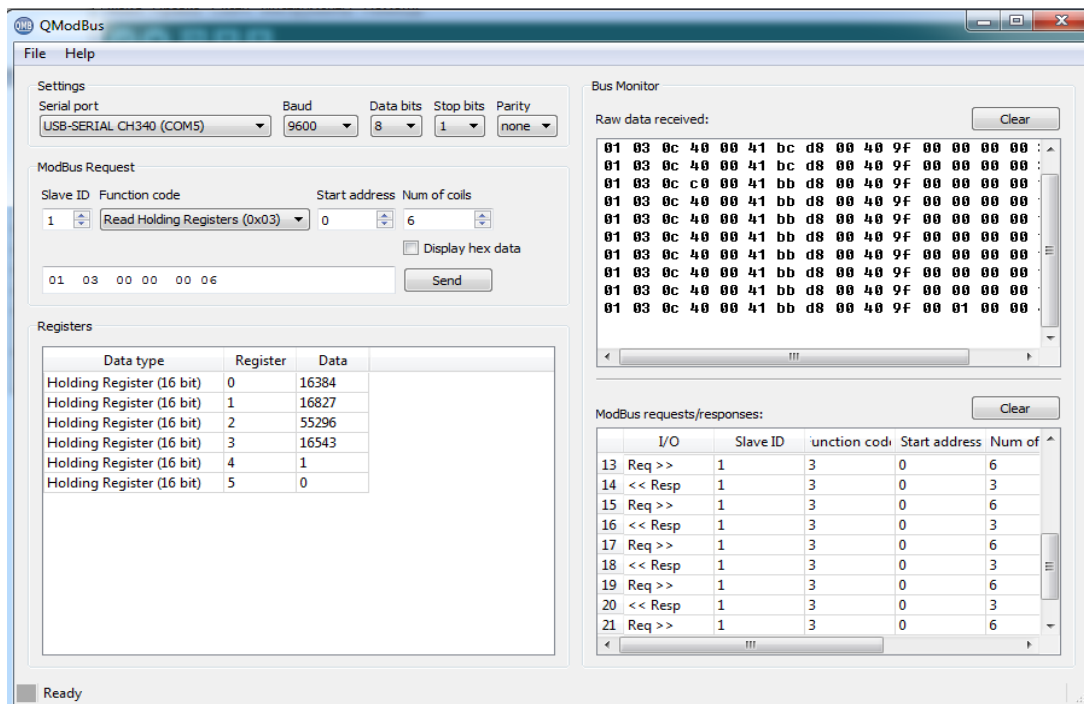
Avvalo, ModBus registrlarini saqlash maqsadini tanlashingiz kerak.

Ro'yxatdan mi	o'tish	Raqam formati	Parametr
0	bitta	suzmoq	Harorat
2			
3	bit	bit	Tugma holati (eng kam
4			
besh	bit	bit	LED holati (eng kam bit)

Eslatib o'taman, ModBus protokolidagi registrlar o'n olti bitli. 6 ta registr chiqdi.

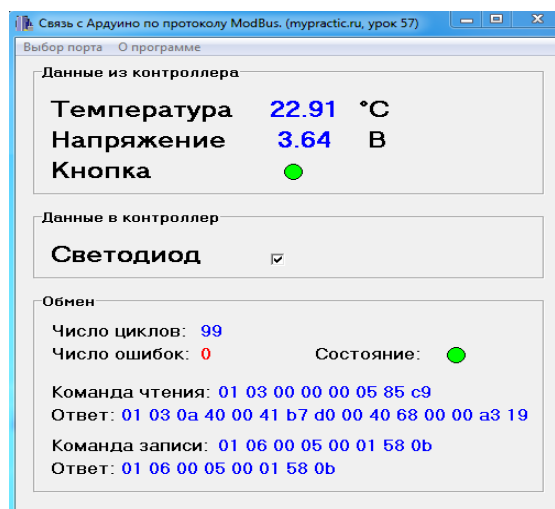
Oldingi eskizni olish, sensorlarni qayta ishlashni asosiy halqaga qo'shish va o'lchangan qiymatlarni saqlash registrlariga yuklash qoladi.

Eskizni ushbu havoladan yuklab olish mumkin (<http://mypractic.ru/?action-rcl=ro'yxatdan+o'ting>) :



15.8-rasm. Kompyuter va Arduino uchun QModBus kompyuter interfeysi .

Men 6 ta saqlash registrlarini o'qib chiqdim va uzatish xatosiz sodir bo'lishiga ishonch hosil qildim, ma'lumotlar o'zgaradi. Faqat tugmalarni tekshirish va LEDni boshqarish bexato bo'lib chiqdi.



15-rasm. Yuqori darajadagi dastur “Arduino bilan ModBus protokoli orqali muloqot qilish ”

Biz raqamli protokoldan foydalanamiz. Shuning uchun sensorlarning o'qishlari ma'nosiz raqamlar to'plamiga o'xshaydi. Va uning yordami bilan men nihoyat mahalliy kontrollerning ishlashini tekshirdim. Hammasi mukammal ishlaydi. Yuqori darajadagi

dasturni havoladan yuklab olishingiz mumkin (<http://mypractic.ru/?action - rcl = ro'yxatdan o'ting>) :

ModBus protokoli va ixtisoslashtirilgan protokolning ishlashini taqqoslash.

Oldingi mashg'ulotda men ModBus protokolini ixtisoslashtirilgan ma'lumotlar almashinuvi protokollari bilan umumiy shaklda, aniq raqamlarsiz solishtirdim. Endi siz bir xil qurilma uchun ikkala protokolning ishlashini hisoblashingiz mumkin.

Maxsus protokol yordamida mahalliy kontroller bilan ma'lumot almashish uchun bizga 15 bayt kerak edi.

Xuddi shu ma'lumotni ModBus protokoli orqali uzatish uchun men 39 baytni hisobladim. Va ramkalar orasidagi ko'proq pauzalar.

```
Команда чтения: 01 03 00 00 00 05 85 c9
Ответ: 01 03 0a 40 00 41 b7 d0 00 40 68 00 00 a3 19
Команда записи: 01 06 00 05 00 01 58 0b
Ответ: 01 06 00 05 00 01 58 0b
```

ModBus tarmog'ining ishlashi ikki baravar past. Bu universal protokolni qo'llash uchun to'lanadigan narx.

Hisobotda quyidagilar bo'lishi kerak:

1. Amaliyot nomlari va ishning maqsadi.
2. Breadboard va tadqiqot tushunchasi.
3. Tadqiqotda foydalaniladigan texnik vositalarning xarakteristikalar.
4. Dastur kodini va kompilyatsiya skrinshotlarini yozish.
5. Bajirilgan vazifa bo'yicha foto / video hisobot.
6. Nazorat savollariga javob yozing (qo'lda)

Nazorat savollari

1. Ushbu loyihaning dasturiy ta'minoti uchun qanday kutubxonalardan foydalandingiz?
2. Loyihani ishlab chiqish uchun zarur bo'lgan apparat komponentlari ro'yxatini tuzing.
3. Ushbu loyihadan qanday maqsadlarda foydalanish mumkin?
4. Datchiklarni Arduino ga ulash sxemasini chizing.
5. Ushbu tadqiqotning sxematik sxemasini tuzing.
6. O'rganish sxemasida datchiklar parametrlariga qanday talablar qo'yiladi?