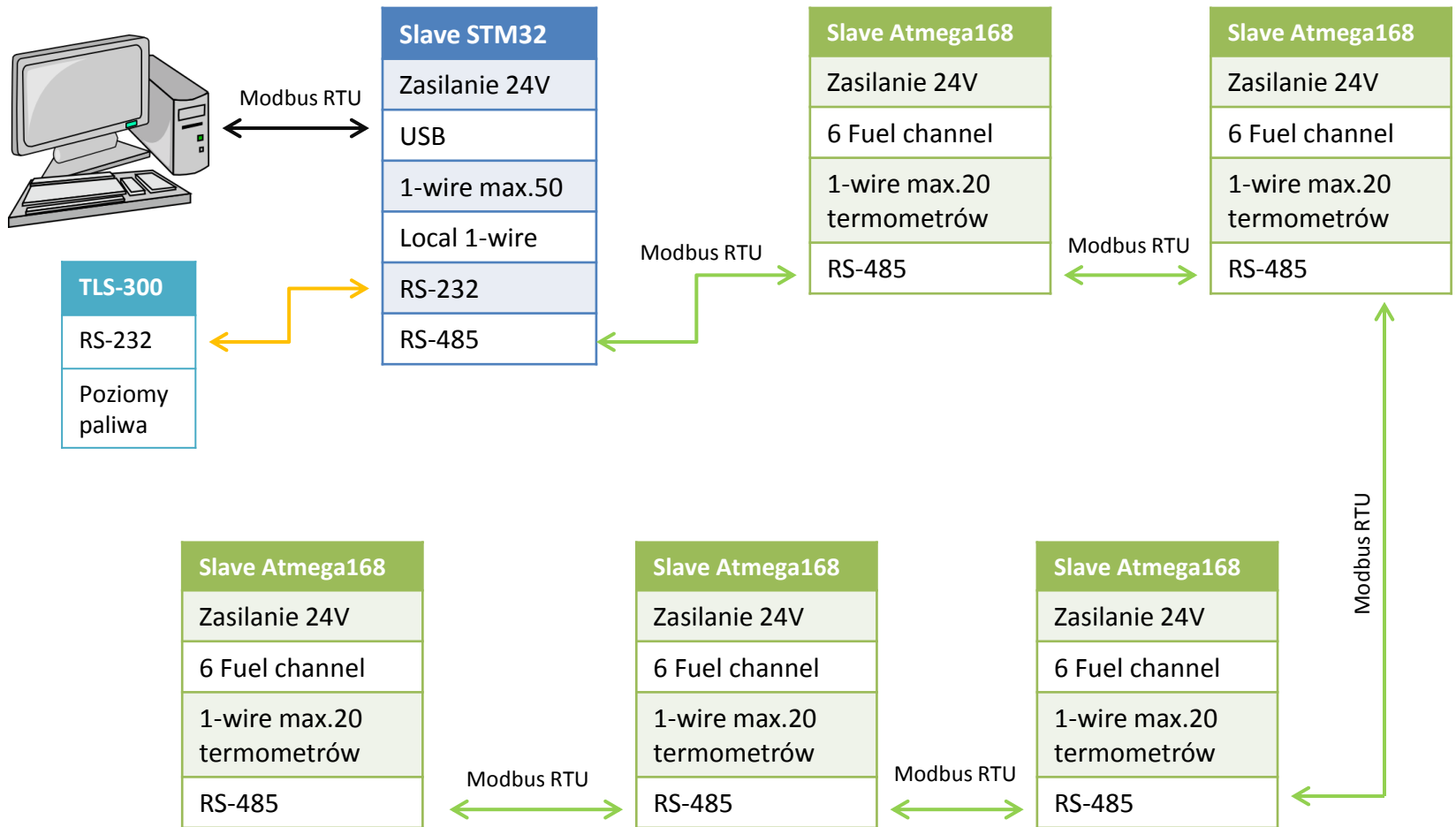


Pomiar zużycia paliwa i temperatur

Strona sprzętowa

Topologia sieci



Moduł SLAVE ATmega168

Propozycja pracy kanałów pomiaru zużycia paliwa

Slave Atmega168

Zasilanie 24V

6 Fuel channel

1-wire max.20
termometrów

RS-485

Adresy cewek (coils) modbusa:

Adres	15	14	13	12	11	10
Uruchamianie kanału pomiaru zużycia paliwa	OIL_CH1 (0/1)	OIL_CH2 (0/1)	OIL_CH3 (0/1)	OIL_CH4 (0/1)	OIL_CH5 (0/1)	OIL_CH6 (0/1)
Adres	65	64	63	62	61	60
Reset przepływu	OIL_CH6 (0/1)	OIL_CH5 (0/1)	OIL_CH4 (0/1)	OIL_CH3 (0/1)	OIL_CH2 (0/1)	OIL_CH1 (0/1)

Aby uzyskać dany przepływ, należy najpierw ustawić 1 w resecie przepływu.

Adresy rejestrów modbusa:

120-131 12 rejestrów	W dwóch rejestrach(32bity) przechowywany średni przepływ. 2 rejestry * 6 kanałów = 12 rejestrów (float CDAB); Średni przepływ z każdego kanału liczony wzorem $X \text{ [ml/min]} = \text{RATIO [ml/imp]} * (\text{ilość_impulsów} * 60000[\text{ms}]) / \text{okres}[\text{ms}]$; RATIO w naszym przypadku to 100 ml na impuls. RATIO dodawane w LabVIEW
140-145 6 rejestrów	W każdym rejestrze przechowywana jest zużycie paliwa (ilość impulsów).

Moduł SLAVE ATmega168

Slave Atmega168
Zasilanie 24V
6 Fuel channel
1-wire max.20 termometrów
RS-485

Propozycja pracy kanałów pomiaru zużycia paliwa cd.

Metoda pomiaru czasu impulsu.

W procesorze uruchomiony Timer sprzętowy, który co 1ms wyzwała przerwanie. W przerwaniu inkrementowana jest zmienna uint32_t czas_glowny.

Przy zboczu narastającym z danego kanału wyzwalane jest przerwanie PCINT.

W przerwaniu realizowany jest następujący kod:

```
If(czas_bufor[kanal]) //jeżeli czas_bufor różny od zera
{
    okres_impulsow[kanal]=czas-czas_bufor[kanal];
    zuzycie_paliwa[kanal]++;
}
```

Czas_bufor[kanal]=czas_glowny;

Przykład:

Warunki początkowe: czas_glowny=150; okres_impulsow=0; czas_bufor=0, impulsy występują co 200 ms

Nr impulsu	1	2	3	4
Okres_impulsow[kanal]	0	200	200	200
Czas_bufor[kanal]	0 -> 150	150 -> 350	350->550	550->750
Czas_glowny	150	350	550	750
zuzycie_paliwa[kanal]	0	1	2	3

Moduł SLAVE ATmega168

Slave Atmega168

Zasilanie 24V

6 Fuel channel

1-wire max.20
termometrów

RS-485

Propozycja pracy kanałów pomiaru zużycia paliwa cd.

Sposób uzyskania przepływu.

W trakcie przerwania PCINT wyzwolonego impulsem oprócz liczenia okresu, zbierana jest ilość impulsów od ostatniego resetu(resetu procesora lub resetu przepływu). W momencie resetu przepływu danego kanału, czyli ustawienia 1 w danej cewce(tabela niebieska kolumna, moment między 4 a 5 impulsem) obliczany jest przepływ od momentu poprzedniego resetu przepływu. Zerowane są natomiast okres impulsów i ilość impulsów.

Sekwencja odczytu przepływu np. OIL_CH1:

1. Ustawienie 1 w adresie cewki 60, czyli reset zbierania średniego przepływu,
2. Odczyt adresu 110 i 111, czyli odczyt obliczonego przepływu.

Warto wspomnieć, że przepływ ma typ float CD AB, czyli aby uzyskać właściwy wynik należy odczyt zapisać do zmiennej typu float w następującej sekwencji: liczba z adresu 111, potem liczba z adresu 110. Można to zrobić na przykład w ten sposób: (float) zmienna=a111, potem zmienna=(zmienna<<16) , potem zmienna|=a110; Gdzie a111 to wartość z adresu 111, a110 to wartość z adresu 110, operator | to lub, operator << to przesunięcie bitowe.

Nr impulsu	1	2	3	4	4	5	6
Ilość impulsów	0	1	2	3	3	0	1
Okres_impulsów[kanał] (SUMA)	0	200	400	600	600	0	200
Reset przepływu (kanał)	0	0	0	0	1	0	0
Przepływ	0	0	0	0	0	300	300
zuzycie_paliwa[ka nał]	0	1	2	3	3	4	5

Warunki początkowe: czas_główny=150; okres_impulsów=0; czas_bufor=0, impulsy występują co 200 ms

Moduł SLAVE ATmega168

Slave Atmega168
Zasilanie 24V
6 Fuel channel
1-wire max.20 termometrów
RS-485

Propozycja pracy kanałów pomiaru zużycia paliwa cd.

Propozycje akwizycji zużycia paliwa.

- Częste odpytywanie modułu w celu uzyskania okresów impulsów i ilości impulsów z każdego kanału. Następnie obróbka tych danych w LabVIEW.
- Jak wyżej z tą różnicą, że zamiast odczytywania okresu, w module obliczane jest chwilowe zużycie [ml/min] i wysyłane razem z ilością impulsów. LabVIEW będzie tylko przemnażać zużycie przez ilość mililitrów na impuls. Istnieje możliwość umieszczenia tego współczynnika w pamięci procesora. Jednak należy pamiętać przy zmianach mierników o modyfikacji tych wartości.
- Rzadsze odpytywanie modułu w celu uzyskania okresów impulsów i ilości impulsów z każdego kanału. Moduł będzie obliczał średni czas trwania impulsu od momentu ostatniego odczytu.
- Jak w pkt.3 z tą różnicą, że zamiast odczytywania okresu, w module obliczane jest chwilowe średnie zużycie [ml/min] i wysyłane razem z ilością impulsów. LabVIEW będzie tylko przemnażać zużycie przez ilość mililitrów na impuls. Istnieje możliwość umieszczenia tego współczynnika w pamięci procesora. Jednak należy pamiętać przy zmianach mierników o modyfikacji tych wartości.

Proszę o dodanie uwag, propozycji, pytań.

Przypuszczam, że częstotliwość impulsów będzie niska w związku z czym uważam, że dwa ostatnie punkty to najlepsza propozycja. Oczywiście wszystko można zmodyfikować wedle uznania.

Moduł SLAVE ATmega168

Propozycja pracy magistrali 1-wire

Slave Atmega168

Zasilanie 24V

6 Fuel channel

1-wire max.20
termometrów

RS-485

Adresy cewek (coils) modbusa:

20-39
20 - bitów

Sygnalizacja błędów termometrów. Np.: Jeżeli na termometrze nr 5 wystąpi błąd to cewka(bit) w adresie 24 będzie miał stan wysoki.

40-59
20 – bitów
(opcja)

Włączanie/ wyłączanie termometrów.

Adresy rejestrów modbusa:

20-39(input
register)
20 rejestrów

Odczyt temperatury. Starszy bajt słowa (rejestru) to część całkowita temperatury ze znakiem, młodszy bajt to miejsce po przecinku.

40-119
(holding
register)
80 rejestrów

Adresy termometrów. Jeden adres termometru zajmuje 4 rejestry(8 bajtów). Tablica ta będzie przechowywana w pamięci EEPROM procesora. Są dwie przyczyny tego rozwiązania. Po pierwsze szkoda SRAM na tak dużo informacji. Po drugie, po zaniku zasilania dane te nadal będą w pamięci.

1(holding
register)

W tym rejestrze przechowywane jest adres slave. Domyślnie jest ustawiony na 1. Po podłączeniu powinno się go zmienić na adres z puli 2-127. Po zapisie nowego adresu nie istnieje możliwość powrotu do wartości 1.

Adres termometru zapisany w rejestrach 40-43 jest pierwszym termometrem. Jego pomiar jest przechowywany w rejestrze 20. Sygnalizacja jego ewentualnego błędu jest przechowywana w cewce 20. Włącznik tego termometru to cewka 40.

Moduł SLAVE ATmega168

Propozycja pracy magistrali 1-wire cd.

Slave Atmega168
Zasilanie 24V
6 Fuel channel
1-wire max.20 termometrów
RS-485

Do uruchamiania sekwencji wykorzystano timer programowy. Sekwencja wygląda następująco. W pierwszej sekundzie wysyłany jest rozkaz do wszystkich termometrów o rozpoczęcie pomiaru. W drugiej sekundzie każdy termometr jest odpytywany, a temperatura zapisywana jest w odpowiednich rejestrach. Czasy te można odpowiednio modyfikować, tak na razie działa prototyp. Do pomiarów proponuję 11-bitową dokładność, co da czas konwersji ok. 350 ms.



Proszę o dodanie uwag, propozycji, pytań.

Moduł SLAVE ATmega168

Informacje na temat RS-485

Slave Atmega168

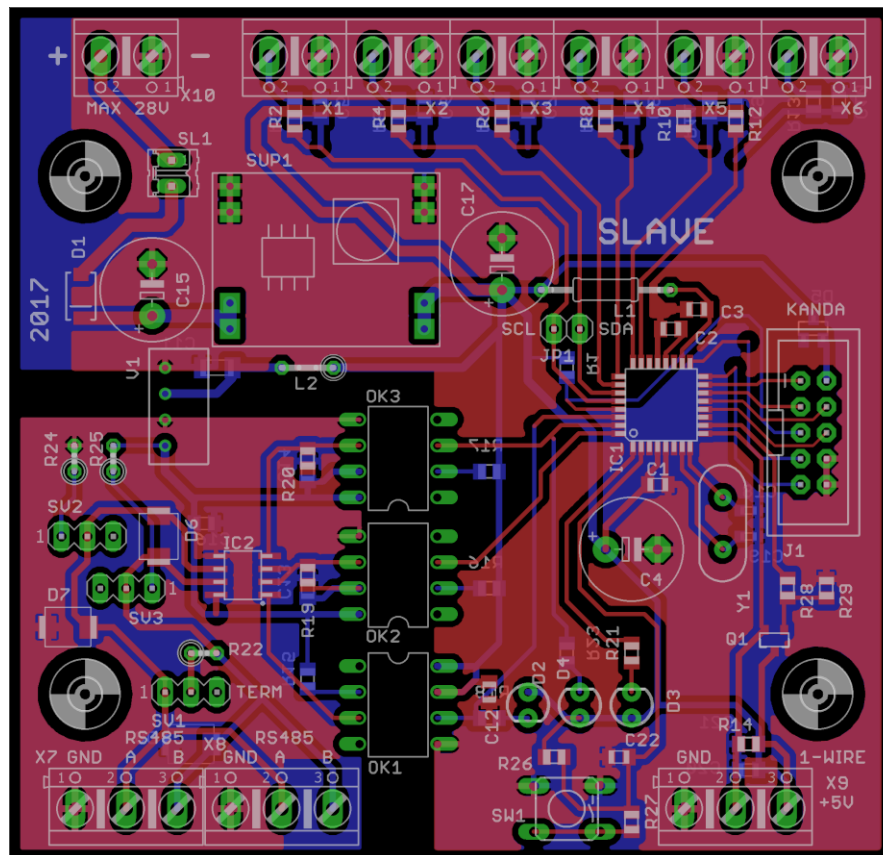
Zasilanie 24V

6 Fuel channel

1-wire max.20
termometrów

RS-485

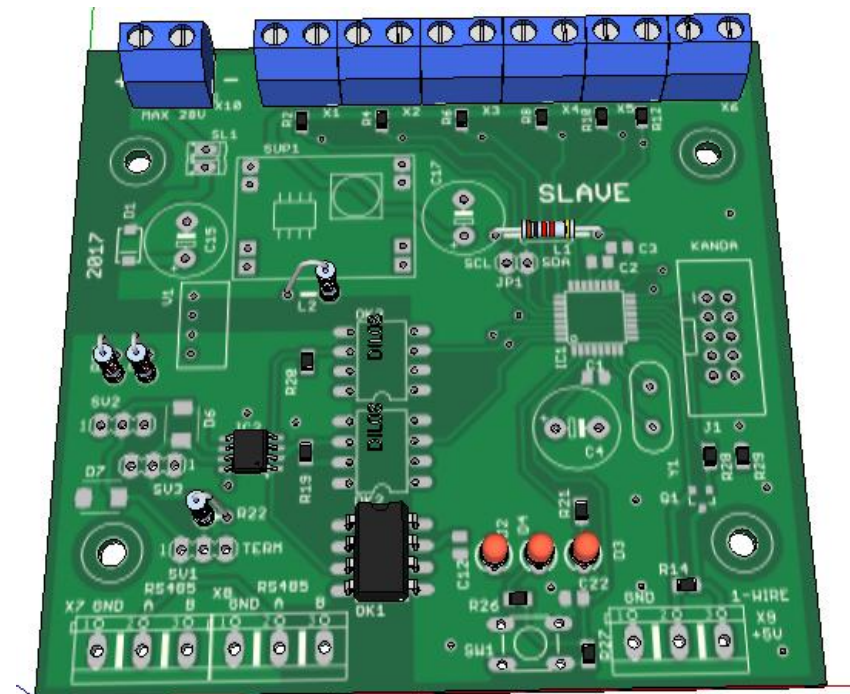
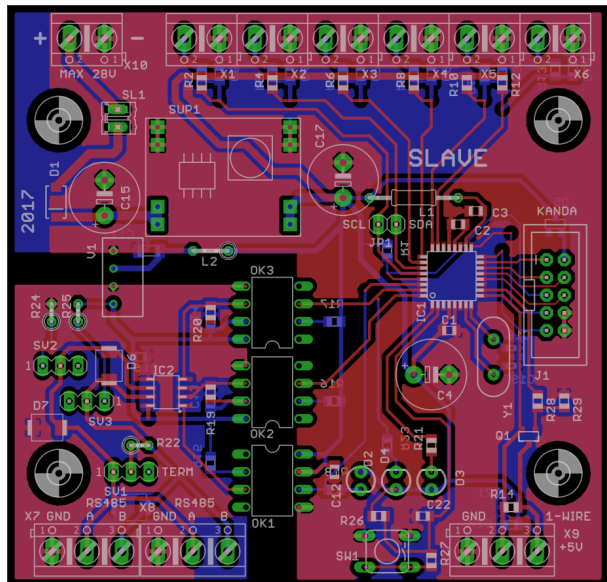
- Separacja galwaniczna,
- Terminator załączany zworką,
- Tryb fail safe załączany zworkami,
- Transile zabezpieczające,
- Domyślny baud rate 38400.



Moduł SLAVE ATmega168

Wizualizacja płytki

Slave Atmega168
Zasilanie 24V
6 Fuel channel
1-wire max.20 termometrów
RS-485



Moduł SLAVE STM32

Slave STM32
Zasilanie 24V
USB
1-wire max.50
Local 1-wire
RS-232
RS-485

Moduł slave STM32 oferuje możliwości podobne do tych z slave'a Atmega168 tylko bez pomiaru zużycia paliwa. Na pokładzie znajduje się więcej portów:

- USB, który pełni rolę konwertera USB <-> RS-485, po USB odpytywany będzie STM32,
- 1-wire do obsługi termometrów,
- Local 1-wire do dodawania nowych termometrów w obiekcie,
- RS-232 do komunikacji z TLS 300,
- RS-485 do komunikacji z pozostałymi modułami.

Moduł SLAVE STM32

Slave STM32

Zasilanie 24V

USB

1-wire max.50

Local 1-wire

RS-232

RS-485

Adresy związane ze stanem modułu STM32

Adresy cewek (coils) modbusa:

3 FRESH_START

Cewka 3 to świeży start. Po każdym resecie procesora, np.: w przypadku zaniku zasilania, ma wartość 1. Powinno się go zawsze sprawdzać i zerować po wgraniu konfiguracji.

Adres slave modułu to niemodyfikowalna wartość 128.

Moduł SLAVE STM32

Slave STM32

Zasilanie 24V

USB

1-wire max.50

Local 1-wire

RS-232

RS-485

Adresy związane z magistralą 1-wire

Adresy cewek (coils) modbusa:

15-64
50 - bitów

Sygnalizacja błędów termometrów. Np.: Jeżeli na termometrze nr 5 wystąpi błąd to cewka(bit) w adresie 19 będzie miał stan wysoki.

70-119
20 – bitów
(opcja)

Włączanie/ wyłączanie termometrów.

Adresy rejestrów modbusa:

20-69(input
register)
50 rejestrów

Odczyt temperatury. Starszy bajt słowa (rejestru) to część całkowita temperatury ze znakiem, młodszy bajt to miejsce po przecinku.

40-239
(holding
register)
200 rejestrów

Adresy termometrów. Jeden adres termometru zajmuje 4 rejestry(8 bajtów). W przypadku tego modułu zawartość nie jest przechowywana w pamięci EEPROM, gdyż moduł takiej nie posiada. Stąd istnieje obowiązek wgrania adresów po każdym wykryciu resetu procesora (sprawdzanie bitu FRESH_START).

Adres termometru zapisany w rejestrach 40-43 jest pierwszym termometrem. Jego pomiar jest przechowywany w rejestrze 20. Sygnalizacja jego ewentualnego błędu jest przechowywana w cewce 15. Włącznik tego termometru to cewka 70.

Moduł SLAVE STM32

Slave STM32

Zasilanie 24V

USB

1-wire max.50

Local 1-wire

RS-232

RS-485

Adresy związane z magistralą Local 1-wire

Adresy cewek (coils) modbusa:

4 CHECK_LOCAL	Wartość 1 uruchamia procedurę odczytu adresu i temperatury z termometru podłączonego do lokalnego złącza 1-wire.
---------------	--

65	Sygnalizacja błędu w trakcie obsługi lokalnego termometru.
----	--

Adresy rejestrów modbusa:

19(input register)	Odczyt lokalnej temperatury. Starszy bajt słowa (rejestru) to część całkowita temperatury ze znakiem, młodszy bajt to miejsce po przecinku.
--------------------	---

15-18 (holding register) 4 rejestry	Adres lokalnego termometru. Jeden adres termometru zajmuje 4 rejestry(8 bajtów).
---	--

Moduł SLAVE STM32

Slave STM32

Zasilanie 24V

USB

1-wire max.50

Local 1-wire

RS-232

RS-485

Adresy związane z obsługą zbiorników paliwa(RS-232)

Adresy cewek (coils) modbusa:

66-68

Sygnalizacja błędu w trakcie obsługi zbiorników.

Adresy rejestrów modbusa:

5 (holding
register)
CHECK_TANK

Zmiana wartości rejestru na 1 rozpoczyna procedurę odczytu danych ze zbiorników paliwa. Za pomocą tego rejestru można śledzić etapy procedury odczytu.
Poniżej wartość rejestru wraz z opisem etapu:
-0 – odczyt wyłączony,
- 1 – wysłanie polecenia do sterownika zbiorników o odczyt zbiornika nr 1,
-2 – oczekiwanie na dane o zbiorniku nr 1,
-3 – przeliczenie odebranych danych zbiornika nr 1 (dane gotowe do odczytu) oraz wysłanie polecenia do sterownika zbiorników o odczyt zbiornika nr 2,
- 4 – oczekiwanie na dane o zbiorniku nr 2,
-5 - przeliczenie odebranych danych zbiornika nr 2 (dane gotowe do odczytu) oraz wysłanie polecenia do sterownika zbiorników o odczyt zbiornika nr 3,
-6 - oczekiwanie na dane o zbiorniku nr 3,
- 7 - przeliczenie odebranych danych zbiornika nr 3 (dane gotowe do odczytu).

70-77(zbiornik1)
78-85(zbiornik2)
86-93(zbiornik3)
(input register)

W dwóch rejestrach(32bity) w formie (float CDAB) przechowywane są:
POJEMNOŚĆ, POJEMNOŚĆ KT, REZERWA, SŁUP.
Dla zbiornika 1 POJEMNOŚĆ KT to rejestry 71-72. Dla zbiornika 3 słup cieczy to rejestry 92-93.

Moduł SLAVE STM32

Slave STM32

Zasilanie 24V

USB

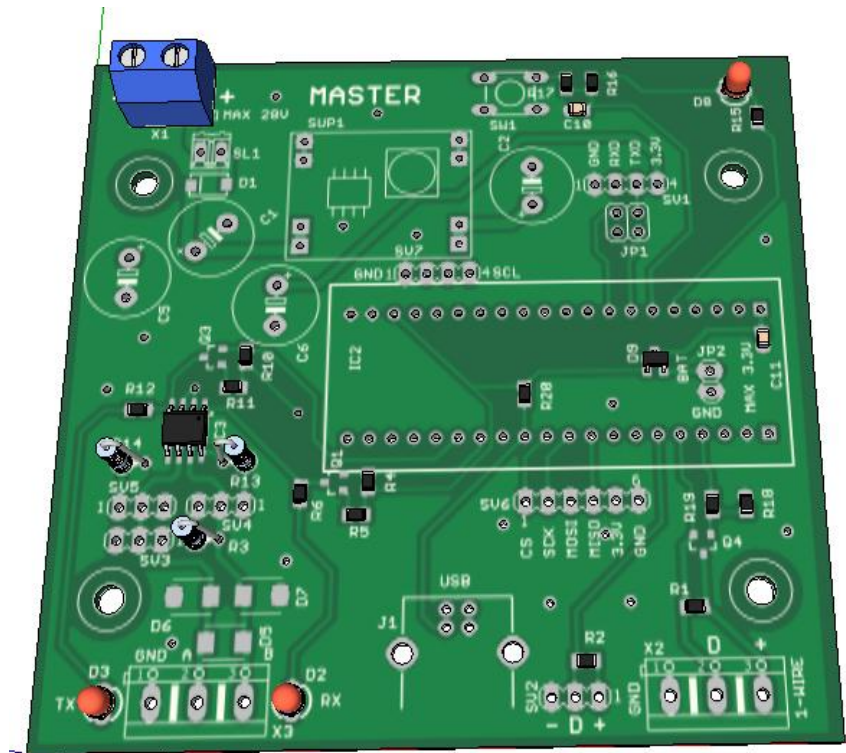
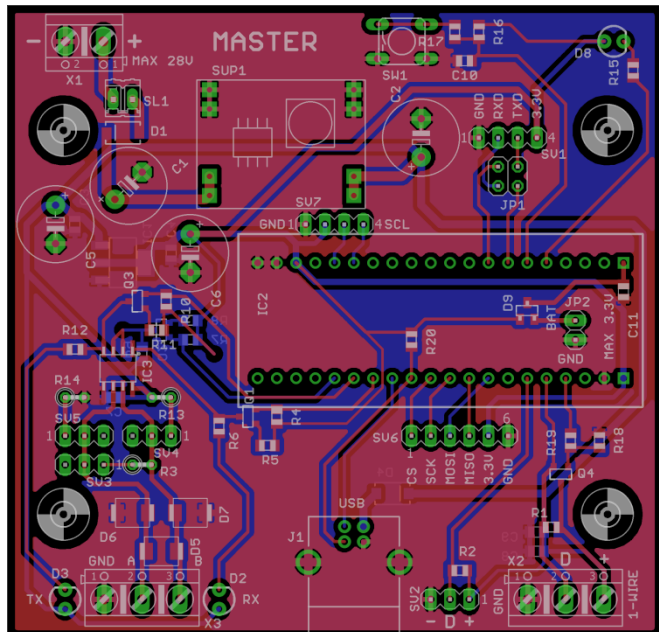
1-wire max.50

Local 1-wire

RS-232

RS-485

Wizualizacja płytki



Propozycja przebiegu konfiguracji modułów ATmega168 w LabVIEW

Ekran Konfiguracja

ID			5			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
OIL_C H6	OIL_CH 5	OIL_ CH4	OIL_CH 3	OIL_ C H2	OIL_ C H1	Lokalna temperatura
TERMOMETRY						
1 <input checked="" type="checkbox"/>	POLE na adres		Opis np.: umiejscowienia(ag regat, pompa)			Przycisk Czytaj
2..20 <input checked="" type="checkbox"/>	POLE na adres		Opis np.: umiejscowienia(ag regat, pompa)			Przycisk Czytaj
Częstotliwość odpytywania			2000 ms			

Przycisk zapisz konfigurację i
prześlij

Po wciśnięciu przycisku „Czytaj” (może być inny tekst) czytany jest termometr umieszczony w porcie LOCAL 1-WIRE modułu STM32, znajdującego się niedaleko komputera. Wtedy LabVIEW wyśle polecenie zmiany cewki(bitu) z 0 na 1. Wtedy wyzwalany jest odczyt adresu i temperatury. Pole adres wtedy wypełni się jakimś adresem. Konfiguracja całego obiektu powinna być gdzieś zapisana do pliku, aby w razie zaniku zasilania przywrócić obiekt do poprzedniego stanu.

To taka luźna propozycja.