

Protokół komunikacyjny

Autor: Marcin Ogórkiewicz

Wstęp

Program służy do obsługi układu mikroprocesorowego, bazującego na płycie STM32 NUCLEO F401RE oraz sondy wodoodpornej zawierającej czujnik temperatury DS18B20. Program obsługuje pomiar z interwałem zdefiniowanym przez użytkownika, domyślnie ustawionym na 60 sekund, w stopniach Celsjusza, przechowywanie oraz wyświetlanie danych temperatury zmierzonych przez czujnik. Ponadto, program pozwala na podpięcie dziewięciu dodatkowych czujników tego samego typu, wyświetlanie ich adresów oraz wybór czujnika do generowania zapisywanych danych.

Wprowadzanie danych

Program przyjmuje dane w postaci znaków ASCII. W celu przesłania znaków „, „”, „;” oraz „\0” należy skorzystać z kodowania przedstawionego poniżej:

Znak	Sposób wprowadzania
!	!!
\0	!1
,	!2
;	!3

Suma kontrolna

Algorytm wybrany przeze mnie do liczenia sumy kontrolnej to:

CRC według standardu CRC16/ARC, z wykorzystaniem wielomianu= 0x8005, wartością początkową= 0x0000, wielkością 4B, odwróceniem bitów wejściowych oraz wyjściowych i wartością reszty=0x0000.

Nadawcy i odbiorcy

Każdy nadawca i odbiorca ma 3 znakowy identyfikator. W przykładowej komunikacji poprzez program, będę posługiwał się następującymi identyfikatorami.

HST – komputer osobisty

STM – mikrokontroler

Tabela komend

Komenda	Działanie
VIEWLAST	Wyświetla ostatnio zapisane w buforze końcowym dane

VIEWR(yyyy-xxxx)	Wyświetla dane zapisane w buforze kołowym o indeksach z zakresu włącznie od yyyy do xxxx włącznie, gdzie x oraz y reprezentują cyfry indeksu w systemie dziesiętnym. Maksymalny możliwy rozmiar zakresu wynosi 10 pozycji
VIEW(yyyy)	Wyświetla rekord w kolejności x od początku buforu (litery x reprezentują cyfry indeksu rekordu w systemie dziesiętnym, VIEW0003 wyświetla rekord trzeci w kolejności od początku)
CHCDEV	Wyświetla adres aktualnie wybranego do pomiarów urządzenia 1-wire
SRCDEV	Wyszukiwanie, wyświetlanie do 9 urządzeń podpiętych do magistrali 1-wire
CHGDEV(yyyyyyyyyyyyyyyy)	Rozpocznij pomiar wartości urządzeniem o podanym adresie 1-wire, wpisanym w miejsce liter „x”
CHCINT	Sprawdź aktualnie ustawiony interwał czasowy pomiarów; czas wyświetlany w sekundach
CHGINT(yyy)	Zmiana interwału czasowego dokonywania pomiarów(w sekundach)

Ramka danych

Minimalna wielkość ramki: 16 znaków

Maksymalna wielkość ramki: 305 znaków

Początek ramki	Nadawca	Odbiorca	Długość danych	Dane	Suma kontrolna	Koniec ramki
1 znak	3 znaki	3 znaki	3 znaki	1 - 290 znaków	4 znaki	1 znak
,	STM lub HST	STM lub HST	Cyfry od 0 do 9 w systemie dziesiętnym (ASCII)	Znaki ASCII, z wykluczeniem „,” oraz „;”	Cyfry od 0 do F w systemie szesnastkowym (ASCII)	;

Poprawne nadawanie komend z komputera osobistego

VIEWLAST:

Początek ramki	Nadawca	Odbiorca	Długość komendy	Dane	Suma kontrolna	Koniec ramki
,	HST	STM	008	VIEWLAST	005D	;

VIEWR(yyyy-xxxx):

Początek ramki	Nadawca	Odbiorca	Długość komendy	Dane	Suma kontrolna	Koniec ramki
,	HST	STM	015	VIEWR(xxxx-yyy)	CRC16/ARC	;

VIEW(xxxx):

Początek ramki	Nadawca	Odbiorca	Długość komendy	Dane	Suma kontrolna	Koniec ramki
,	HST	STM	010	VIEW(xxxx)	CRC16/ARC	;

CHCDEV:

Początek ramki	Nadawca	Odbiorca	Długość komendy	Dane	Suma kontrolna	Koniec ramki
,	HST	STM	006	CHCDEV	C209	;

SRCDEV:

Początek ramki	Nadawca	Odbiorca	Długość komendy	Dane	Suma kontrolna	Koniec ramki
,	HST	STM	006	SRCDEV	9052	;

CHGDEV(xxxxxxxxxxxxxxxxxx):

Początek ramki	Nadawca	Odbiorca	Długość komendy	Dane	Suma kontrolna	Koniec ramki
,	HST	STM	024	CHGDEV(xxxxxxxxxxxxxxxxxx)	CRC16/ARC	;

CHCINT:

Początek ramki	Nadawca	Odbiorca	Długość komendy	Dane	Suma kontrolna	Koniec ramki
,	HST	STM	006	CHCINT	F01E	;

CHGINT(xxx):

Początek ramki	Nadawca	Odbiorca	Długość komendy	Dane	Suma kontrolna	Koniec ramki
,	HST	STM	011	CHGINT(xxx)	CRC16/ARC	;

Odbiór danych:

1. Po odebraniu przez STM komendy VIEWLAST wysyłana jest przez niego pojedyncza ramka danych z temperaturą do komputera HST.
2. Po odebraniu przez STM komendy VIEWR(yyyy-xxxx) wysłana jest przez niego ramka danych z temperaturami, zawartymi w zakresie zdefiniowanym przez użytkownika, do komputera HST.
3. Po odebraniu przez STM komendy VIEW(yyyy) wysyłana jest przez niego pojedyncza ramka danych z temperaturą, z odpowiedniej pozycji w buforze, podanej przez użytkownika, do komputera HST.
4. Po odebraniu przez STM komendy CHCDEV, STM wyświetla adres 1-wire aktualnie używanego czujnika
5. Po odebraniu przez STM komendy SRCDEV, STM wyświetla adresy 1-wire wszystkich urządzeń podłączonych do magistrali do komputera HST.
6. Po odebraniu przez STM komendy CHGDEV(yyyyyyyyyyyyyyyy), STM zmienia używany czujnik na ten o podanym, zamiast liter „x”, adresie, po czym wysyła komunikat zwrotny „SUCCESS”.
7. Po odebraniu przez STM komendy CHCINT, STM wyświetla aktualnie zdefiniowany interwał pomiarowy w sekundach.
8. Po odebraniu przez STM komendy CHGINT(yyy), STM ustawia interwał pomiarowy w sekundach na ten podany przez użytkownika, w miejscu liter „x”, po czym wysyła komunikat zwrotny „SUCCESS”.

Odpowiedź STM na komendy VIEWLAST, VIEWR(yyyy-xxxx), VIEW(yyyy)

Początek ramki	Nadawca	Odbiorca	Długość danych	Dane	Suma kontrolna	Koniec ramki
,	STM	HST	028 - 290	i tt.tt gg:minmin dd.mm.rrrr	CRC16/ARC	;

Struktura danych temperatury

i[spacja]tt.tt[spacja]gg:minmin[spacja]dd.mm.rrrr[enter]

i-indeks

t-znak temperatury

g-znak godziny

min-znak minuty

d-znak dnia

m-znak miesiąca

r-znak roku

Odpowiedź STM na komendy CHCDEV, SRCDEV

Początek ramki	Nadawca	Odbiorca	Długość danych	Dane	Suma kontrolna	Koniec ramki
,	STM	HST	025 - 234	i xx xx xx xx xx xx xx xx	CRC16/ARC	;

Struktura danych dotyczących adresów 1 wire

i[spacja]xx[spacja]xx[spacja]xx[spacja]xx[spacja]xx[spacja]xx[spacja]xx[enter]

i - indeks

x-znak ASCII odpowiadający cyfrze w systemie szesnastkowym adresu

Odpowiedź STM na komendy CHGDEV, CHGINT

Początek ramki	Nadawca	Odbiorca	Długość danych	Dane	Suma kontrolna	Koniec ramki
,	STM	HST	007	SUCCESS	5352	;

Odpowiedź STM na komendy CHCINT

Początek ramki	Nadawca	Odbiorca	Długość danych	Dane	Suma kontrolna	Koniec ramki
,	STM	HST	001 - 003	xxx	CRC16/ARC	;

W miejscu liter "x" znajduje się wielkość interwału pomiarowego w sekundach.

Obsługa błędów:

1. Kilka znaków początku ramki pod rząd – jeżeli zostanie wykrytych kilka znaków początku ramki pod rząd, przetwarzanie ramki powinno zacząć się od ostatniego z tych znaków, reszta powinna zostać zignorowana.
2. Wykrycie znaku początku ramki bez wykrycia znaku końca ramki – ramka nie zostaje znaleziona.
3. Kilka znaków końca ramki pod rząd – pierwszy znak końca ramki w ciągu takich znaków powinien zostać wzięty pod uwagę, a reszta zignorowana.
4. Znak początku ramki po poprzednim wykryciu początku ramki (np. w polu Dane) – jeżeli znak początku ramki zostanie wykryty podczas procesu przetwarzania ramki, proces przetwarzania powinien zacząć się od początku.
5. Nieprawidłowy odbiorca – jeżeli konstrukcja ramki jest prawidłowa, a odbiorca nie wskazuje na to urządzenie, ramka powinna zostać zignorowana.
6. Długość komendy = 0 – Jeżeli odbiorca jest prawidłowy, powinien zostać wysłany odpowiedni komunikat.
7. Długość komendy != ilość znaków w polu Dane – należy zwrócić odpowiedni komunikat.
8. Nieprawidłowa suma kontrolna – należy zwrócić odpowiedni komunikat.
9. Nieprawidłowa komenda – należy zwrócić odpowiedni komunikat.
10. Brak Nadawcy/Odbiorcy/Sumy Kontrolnej/Długości Komendy – ramka zostaje zignorowana.
11. Niepoprawny zakres, indeks bądź interwał – zwrócenie komunikatu o błędnej komendzie

Tabela komunikatów zwrotnych

Komunikat	Znaczenie
CMDLEN0	Długość otrzymanej komendy = 0
CMDLENINC	Długość otrzymanej komendy niezgodna z ilością znaków w polu dane
CCHSUMINC	Nieprawidłowa suma kontrolna
CMDINC	Nieprawidłowa komenda
SUCCESS	Komenda zmiany interwału wykonana poprawnie

Przykładowa odpowiedź STM na błędną komendę

Początek ramki	Nadawca	Odbiorca	Długość danych	Dane	Suma kontrolna	Koniec ramki
,	STM	HST	007	CMDLEN0	E7BF	;

Zasada działania

Połączenie czujników i płytki

Linie, którą będziemy obsługiwać 1-wire, należy podłączyć do gniazda na płycie oznaczonego vcc 3.3V rezystorem pull up (np. 5 kΩ). Żółty kabelek sondy podłączamy do linii 1-wire, czarny do masy, a czerwony do vcc 3.3V. Piny płytki i czujników podłączonych do 1-wire ustawiamy na tryb open-drain.

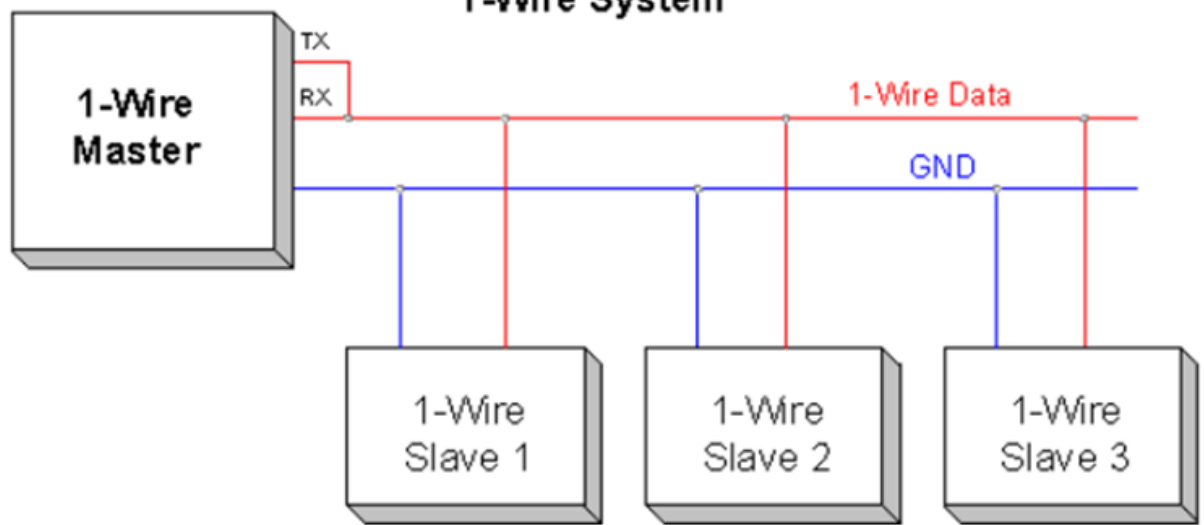
W projekcie wykorzystano interfejs USART (Universal Synchronous/Asynchronous Receiver/Transmitter) do komunikacji z innymi urządzeniami. Interfejs USART umożliwia nie tylko asynchroniczną, ale także synchroniczną transmisję danych. Używany jest do komunikacji z komputerem oraz innymi mikrokontrolerami, umożliwiając monitorowanie temperatury lub przesyłanie danych do innych systemów. Parametry połączenia przez interfejs USART: 115200, 8, N, 1, brak kontroli przepływu.

Projekt korzysta z interfejsu 1-Wire do komunikacji z cyfrowymi czujnikami temperatury DS18B20. 1-Wire wykorzystuje jedną linię danych, gdzie stan wysoki jest stanem bezczynnym. W przypadku DS18B20, każdy czujnik posiada unikalny 64-bitowy identyfikator, co umożliwia obsługę wielu czujników na jednej magistrali 1-Wire.

Komunikacja 1-Wire opiera się na zasadzie impulsów czasowych, gdzie zarówno 0, jak i 1 są reprezentowane przez różnice w czasie trwania impulsów. Mikrokontroler STM32 NUCLEO F401RE, korzystając z magistrali 1-Wire, może wysyłać komendy do czujników DS18B20, inicjować pomiary temperatury i odczytywać dane pomiarowe.

Schemat połączenia:

Figure 1. OneWire Block Diagram
1-Wire System



Wyszukiwanie i wybór czujnika

Każdy czujnik DS18B20 posiada unikalny adres o długości 64 bitów. Algorytm wyszukiwania urządzeń 1-Wire opiera się na drzewie binarnym, gdzie odnogi są śledzone, aż do momentu znalezienia numeru ROM urządzenia lub inaczej liścia. Następnie kolejne wyszukiwania przechodzą ścieżkami odnóg, aż wszystkie obecne liście zostaną odkryte. Algorytm rozpoczyna się od zresetowania urządzeń na magistrali 1-Wire za pomocą sekwencji sygnałów resetu i obecności. Jeśli to się powiedzie, wysyłane jest jednobajtowe polecenie wyszukiwania. Polecenie to przygotowuje urządzenia 1-Wire do rozpoczęcia wyszukiwania. Istnieją dwa rodzaje poleceń wyszukiwania. Normalne polecenie wyszukiwania (F0 hex) wykonuje wyszukiwanie ze wszystkimi urządzeniami uczestniczącymi. Polecenie wyszukiwania alarmu lub warunkowe (EC hex) wykonuje wyszukiwanie tylko z urządzeniami znajdującymi się w jakimś stanie alarmowym. Po poleceniu wyszukiwania rozpoczyna się właściwe wyszukiwanie, gdzie wszystkie uczestniczące urządzenia jednocześnie wysyłają pierwszy bit (najmniej znaczący) swojego numeru ROM. Master 1-Wire inicjuje każdy bit, niezależnie od tego, czy jest to dane do odczytu, czy do zapisu do urządzeń podrzędnych. Ze względu na charakterystykę 1-Wire, gdy wszystkie urządzenia odpowiadają jednocześnie, uzyskuje się logiczne AND wystanych bitów. Po wystaniu przez urządzenia pierwszego bitu ich numeru ROM, master inicjuje kolejny bit, a urządzenia wysyłają wtedy dopełnienie pierwszego bitu. Na podstawie tych dwóch bitów można uzyskać informacje na temat pierwszego bitu w numerach ROM uczestniczących urządzeń. Zgodnie z algorytmem wyszukiwania master 1-Wire musi następnie wysłać bit z powrotem do uczestniczących urządzeń. Jeśli uczestniczące urządzenie ma tę wartość bitu, kontynuuje uczestnictwo. Jeśli nie ma tej wartości bitu, przechodzi w stan oczekiwania do wykrycia kolejnego resetu 1-Wire. Ten wzorzec "czytaj dwa bity" i "zapisuj jeden bit" jest następnie powtarzany dla pozostałych 63 bitów numeru ROM. W ten sposób algorytm wyszukiwania zmusza wszystkie urządzenia, z wyjątkiem jednego, do przejścia w stan oczekiwania. Pod koniec jednego przejścia znany jest numer ROM tego ostatniego urządzenia. W kolejnych przejściach wyszukiwania podjęta zostaje inna ścieżka (lub odnoga), aby znaleźć inne numery ROM urządzeń. Jeśli wszystkie uczestniczące urządzenia mają tę samą wartość w pozycji bitu, istnieje tylko jedno wyjście dla ścieżki.

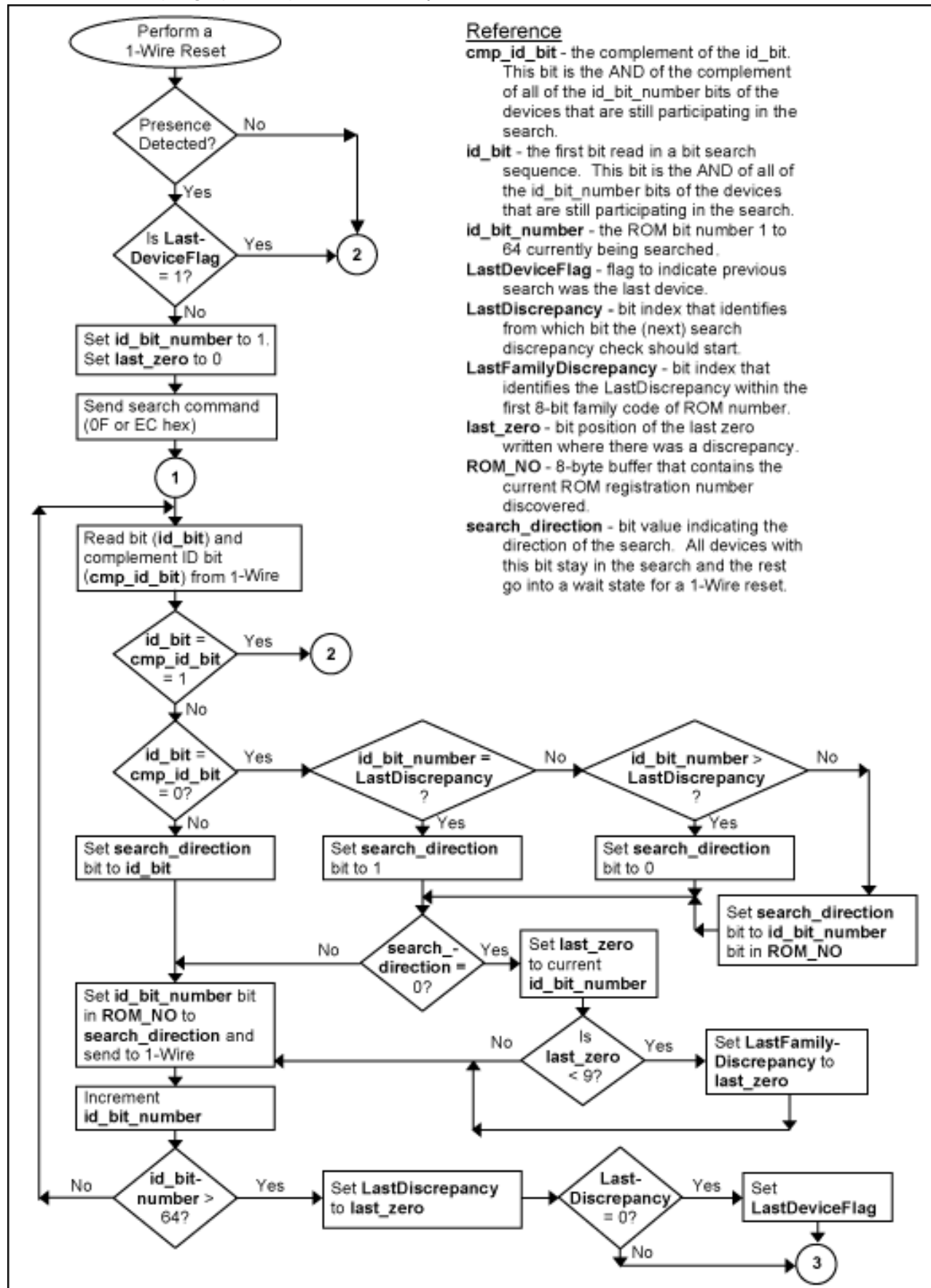
Warunek, w którym żadne urządzenia nie uczestniczą, to nietypowa sytuacja, która może się pojawić, jeśli urządzenie, które jest odkrywane, zostanie usunięte z 1-Wire podczas wyszukiwania. Jeśli taka sytuacja wystąpi, wyszukiwanie powinno być zakończone, a nowe wyszukiwanie można by było rozpocząć od resetu 1-Wire. Niezgodność, czyli warunek, w którym w pozycji bitu występują zarówno 0, jak i 1, jest kluczem do znajdowania urządzeń w kolejnych operacjach wyszukiwania. Algorytm wyszukiwania określa, że w pierwszym przejściu, gdy występuje niezgodność (bit/dopełnienie = 0/0), należy podjąć ścieżkę '0'. Pozycja bitu dla ostatniej niezgodności jest rejestrowana do użycia w następnym wyszukiwaniu.

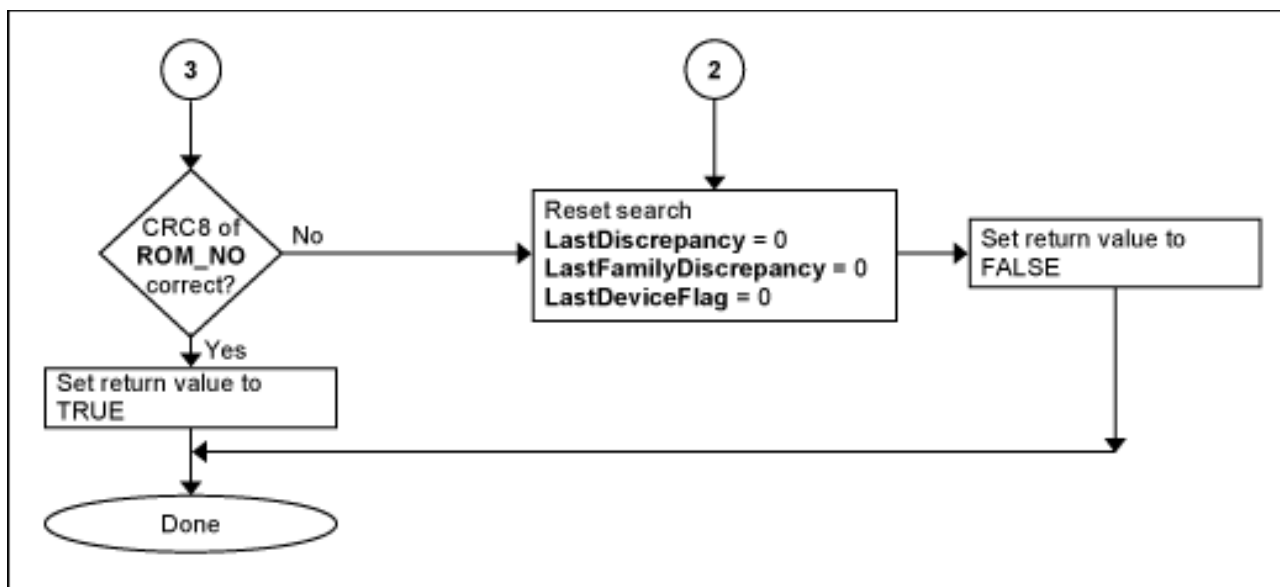
Algorytm wyszukiwania śledzi również ostatnią niezgodność występującą w pierwszych ośmiu bitach algorytmu. Pierwsze osiem bitów 64-bitowego numeru rejestracyjnego to kod rodziny. W rezultacie urządzenia odkryte podczas wyszukiwania są grupowane według rodzajów rodziny. Ostatnią niezgodność w obrębie tego kodu rodziny można użyć do selektywnego pomijania całych grup urządzeń 1-Wire.

64-bitowy numer ROM zawiera również 8-bitową sumę kontrolną cykliczną (CRC). Ta wartość CRC jest weryfikowana, aby zapewnić, że są odkrywane tylko poprawne numery ROM. Istnieją dwa podstawowe rodzaje operacji, które można wykonać za pomocą algorytmu wyszukiwania, manipulując wartościami rejestrów LastDiscrepancy, LastFamilyDiscrepancy, LastDeviceFlag i ROM_NO. Te operacje dotyczą podstawowego odkrywania numerów ROM urządzeń 1-Wire.

Operacja 'FIRST' polega na wyszukiwaniu na 1-Wire pierwszego urządzenia. Wykonywane jest to przez ustawienie wartości LastDiscrepancy, LastFamilyDiscrepancy i LastDeviceFlag na zero, a następnie wykonanie wyszukiwania. Następnie uzyskany numer ROM można odczytać z rejestru ROM_NO. Jeśli na 1-Wire nie ma urządzeń, sekwencja resetowania nie wykrywa obecności, a wyszukiwanie jest przerywane. Następną Operacją 'NEXT' polega na wyszukiwaniu na 1-Wire następnego urządzenia. Wyszukiwanie to zazwyczaj jest wykonywane po operacji 'FIRST' lub innej operacji 'NEXT'. Wykonywane jest to przez pozostawienie stanu bez zmian w porównaniu do poprzedniego wyszukiwania i wykonanie kolejnego wyszukiwania. Następnie uzyskany numer ROM można odczytać z rejestru ROM_NO. Jeśli poprzednie wyszukiwanie było ostatnim urządzeniem na 1-Wire, wynik jest FALSE, a warunek jest ustawiony do wykonania 'FIRST' przy następnym wywołaniu algorytmu wyszukiwania.

Schemat blokowy algorytmu wyszukiwania urządzeń 1-wire:





Konfiguracja czujnika

Przed przystąpieniem do pomiarów należy zdefiniować timer, który będzie pilnował interwału pomiarowego. Program jest przystosowany na domyślnej rozdzielczości danych temperatury, jednakże w razie potrzeby można ją zmniejszyć do dziewięciu bitów. Aby tego dokonać należy zmienić wartość czwartego bajtu pamięci notatnikowej (licząc od najmłodszego bitu); stanowi on rejestr konfiguracji. Poniżej znajduje się tabelka z możliwymi konfiguracjami temperatury.

Bajt 4

0	R1	R0	1	1	1	1	1
MSb			LSb				

Możliwe konfiguracje

R1	R0	Thermometer Resolution	Max Conversion Time
0	0	9 bit	93.75 ms ($t_{conv}/8$)
0	1	10 bit	187.5 ms ($t_{conv}/4$)
1	0	11 bit	375 ms ($t_{conv}/2$)
1	1	12 bit	750 ms (t_{conv})

Pamięć notatnika

SCRATCHPAD		BYTE
TEMPERATURE LSB		0
TEMPERATURE MSB		1
TH/USER BYTE 1		2
TL/USER BYTE 2		3
CONFIG		4
RESERVED		5
RESERVED		6
RESERVED		7
CRC		8

Pomiar i odczyt temperatury

Aby dokonać pomiaru temperatury, należy obsłużyć funkcję, która po odmierzeniu przez timer odpowiedniej ilości czasu wykona polecenie konwersji „44h”, po czym aby dane zostały pobrane do STM32, wykona polecenie odczytu pamięci notatnika (scratchpad) „BEh”. Dane temperatury znajdują się na pierwszych 4 bajtach pamięci. Bajt 0 przechowuje najmniej znaczący bit. Bajt 1 przechowuje najbardziej znaczący bit, który decyduje o tym, czy temperatura jest dodatnia(0), czy ujemna(1). Bajt 2 przechowuje

część całkowitą pomiaru. Bajt 3 przechowuje część dziesiętną temperatury. Aby przekształcić temperaturę na format do przechowywania w buforze i wyświetlania dla użytkownika, trzeba dokonać sumowania wartości bajtów 2 i 3 i ewentualnie przemnożyć ją przez -1, jeżeli wartość bajtu 2 wynosi 1.

Schemat przesyłania danych przez 1-wire:

Figure 2. OneWire Timing Diagram

