

Knihovna CFoxLib

TXV 003 71.01
první vydání
červen 2012
změny vyhrazeny

Historie změn

Datum	Vydání	Popis změn
Červen 2012	1	První vydání, popis odpovídá CFoxLib_v10

OBSAH

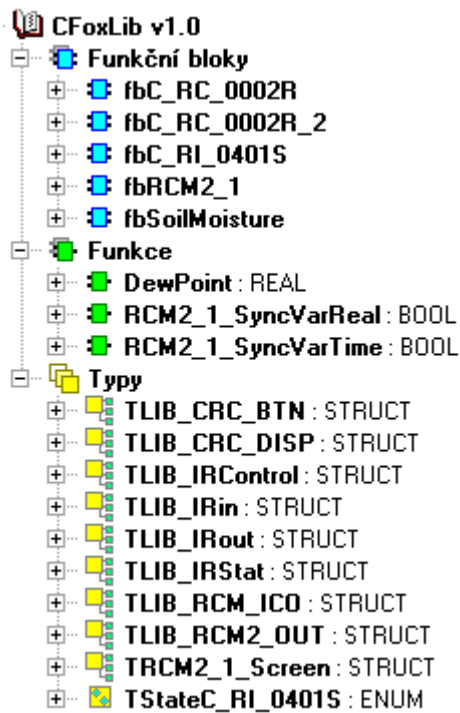
1 Úvod	3
2 Datové typy	4
3 Konstanty	5
4 Globální proměnné	5
5 Funkce	5
5.1 Funkce DewPoint	6
5.2 Funkce RCM2_1_SyncVarReal	7
5.3 Funkce RCM2_1_SyncVarTime	8
6 Funkční bloky	9
6.1 Funkční blok fbC_RC_0002R	10
6.2 Funkční blok fbC_RC_0002R_2	12
6.3 Funkční blok fbC_RI_0401S	15
6.4 Funkční blok fbRCM2_1	22
6.5 Funkční blok fbSoilMoisture	30

1 ÚVOD

TXV 003 71.01

Knihovna CFoxLib je standardně dodávána jako součást programovacího prostředí Mosaic. Knihovna obsahuje funkce a funkční bloky podporující možnosti modulů v síti CFox.

Následující obrázek ukazuje strukturu knihovny CFoxLib v prostředí Mosaic



Pokud chceme funkce z knihovny CFoxLib použít v aplikačním programu PLC, je třeba nejprve přidat tuto knihovnu do projektu. Knihovna je dodávána jako součást instalace prostředí Mosaic od verze 2012.3.

2 DATOVÉ TYPY

V knihovně CFoxLib jsou definovány následující datové typy:

<i>Identifikátor</i>	<i>Typ</i>	<i>Význam</i>
<i>TStateC_RI_0401S</i>	ENUM	Stavy bloku pro IR ovládání pro C-RI-0401S (popis jednotlivých hodnot viz funkční blok <i>fbC_RI_0401S</i>)
<i>TRCM2_1_Screen</i>	STRUCT	Struktura popisující chování obrazovky zobrazené na modulu RCM2-1 (více viz funkční blok <i>fbRCM2_1</i>)

Dále jsou znovu definovány následující struktury modulů. Tyto struktury se využívají pro vstupně výstupní proměnné sloužící k předání dat jednotlivých modulů.

<i>Identifikátor</i>	<i>Typ</i>	<i>Význam</i>
<i>TLIB_CRC_BTN</i>	STRUCT	Struktura tlačítek C-RC-0002R
<i>TLIB_CRC_DISP</i>	STRUCT	Struktura displeje C-RC-0002R
<i>TLIB_IRControl</i>	STRUCT	Řídící byte IR vysílače C-RI-0401S
<i>TLIB_IRin</i>	STRUCT	Vstupní struktura C-RI-0401S
<i>TLIB_IRout</i>	STRUCT	Výstupní struktura C-RI-0401S
<i>TLIB_IRStat</i>	STRUCT	Stavový byte IR přijímače C-RI-0401S
<i>TLIB_RCM_ICO</i>	STRUCT	Struktura viditelnosti symbolů na displeji RCM2-1
<i>TLIB_RCM2_OUT</i>	STRUCT	Výstupní struktura RCM2-1

3 KONSTANTY

V knihovně CFoxLib nejsou definovány žádné konstanty

4 GLOBÁLNÍ PROMĚNNÉ

V knihovně CFoxLib nejsou definovány žádné globální proměnné.

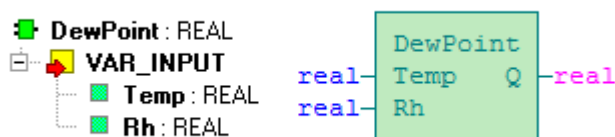
5 FUNKCE

Knihovna CFoxLib obsahuje následující funkce:

<i>Funkce</i>	<i>Popis</i>
<i>DewPoint</i>	Vrací rosný bod [°C] na základě teploty v prostoru [°C] a relativní vlhkosti [%] (C-AQ-0004R)
<i>RCM2_1_SyncVarReal</i>	Synchronizace editovatelné REAL proměnné na obrazovce RCM2-1, vrací TRUE pokud je obrazovka RCM2-1 nastavena pro zobrazení reálných čísel
<i>RCM2_1_SyncVarTime</i>	Synchronizace editovatelné TIME proměnné na obrazovce RCM2-1, vrací TRUE pokud je obrazovka RCM2-1 nastavena pro zobrazení času

5.1 Funkce DewPoint

Knihovna : CFoxLib



Funkce *Dewpoint* vrací rosný bod ve stupních Celsia na základě teploty v prostoru a relativní vlhkosti v procentech. Relativní vlhkost lze získat pomocí modulu C-AQ-0004R.

Rosný bod je vypočítán dle následujícího vztahu a zaokrouhlen na celé stupně:

$$DewPoint = \frac{5745,31}{\frac{5745,31}{Temp + 273,15} - \ln\left(\frac{Rh}{100,0}\right)} - 273,15$$

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Temp</i>	REAL	Teplota v prostoru ve stupních Celsia
	<i>Rh</i>	REAL	Relativní vlhkost v procentech
DewPoint			
	<i>Návratová hodnota</i>	REAL	Rosný bod ve stupních Celsia

Příklad programu s voláním funkce *DewPoint* :

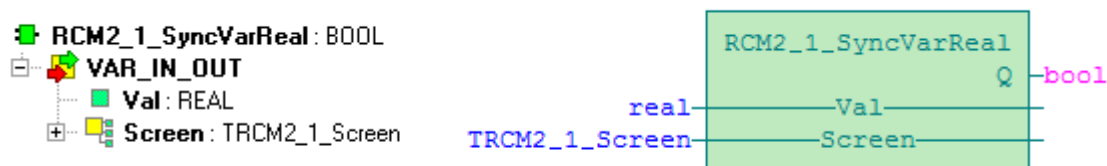
```
PROGRAM prgDewPointExample
  VAR
    DewPointTemp : REAL;
  END_VAR

  DewPointTemp := DewPoint(Temp := C_RC_0002R_IN.eTHERM,
                           Rh := C_AQ_0004R_IN.RH);

END_PROGRAM
```

5.2 Funkce RCM2_1_SyncVarReal




Knihovna : CFoxLib



Funkce *RCM2_1_SyncVarReal* je určena k použití s blokem *fbRCM2_1*. Funkce umožňuje synchronizaci editovatelné proměnné typu REAL na obrazovce modulu RCM2-1. Synchronizace je prováděna tak, aby bylo možné měnit proměnnou z více míst. Toho je dosaženo tak, že se do proměnné zapisuje jen v okamžiku ukončení editace.

Funkce vrací hodnotu TRUE v případě, že je synchronizovaná proměnná na obrazovce nastavené pro zobrazení hodnot typu REAL.

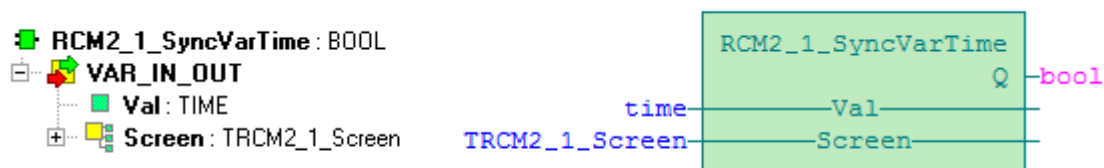
Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_IN_OUT			
	<i>Val</i>	REAL	Synchronizovaná REAL proměnná
	<i>Screen</i>	TRCM2_1_Screen	Struktura popisující obrazovku, na které se proměnná edituje
SyncVarReal			
	<i>Návratová hodnota</i>	BOOL	TRUE pokud je obrazovka RCM2-1 nastavena pro zobrazení reálných čísel

Příklad programu s voláním funkce *RCM2_1_SyncVarReal* je součástí příkladu k bloku *fbRCM2_1*.

5.3 Funkce RCM2_1_SyncVarTime

Knihovna : CFoxLib



Funkce *RCM2_1_SyncVarTime* je určena k použití s blokem *fbRCM2_1*. Funkce umožňuje synchronizaci editovatelné proměnné typu TIME na obrazovce modulu RCM2-1. Synchronizace je prováděna tak, aby bylo možné měnit proměnnou z více míst. Toho je dosaženo tak, že se do proměnné zapisuje jen v okamžiku ukončení editace.

Funkce vrací hodnotu TRUE v případě, že je synchronizovaná proměnná na obrazovce nastavené pro zobrazení hodnot typu TIME.

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_IN_OUT			
	<i>Val</i>	TIME	Synchronizovaná TIME proměnná
	<i>Screen</i>	TRCM2_1_Screen	Struktura popisující obrazovku, na které se proměnná edituje
SyncVarTime			
	<i>Návratová hodnota</i>	BOOL	TRUE pokud je obrazovka RCM2-1 nastavena pro zobrazení času

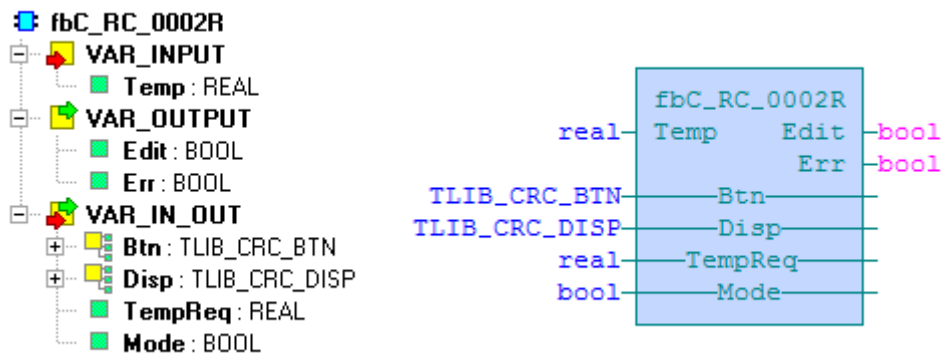
Příklad programu s voláním funkce *RCM2_1_SyncVarTime* je součástí příkladu k bloku *fbRCM2_1*.

6 FUNKČNÍ BLOKY

V knihovně CFoxLib jsou definovány následující funkční bloky:

<i>Funkční blok</i>	<i>Popis</i>
<i>fbC_RC_0002R</i>	Zobrazení teploty, editace žádané teploty a přepínání režimu pro modul C-RC-0002R
<i>fbC_RC_0002R_2</i>	Zobrazení teploty, editace dvou žádaných teplot a přepínání režimu pro modul C-RC-0002R
<i>fbC_RI_0401S</i>	Posílání, příjem a učení IR kódu pro C-RI-0401S
<i>fbRCM2_1</i>	Jednoduché řízení RCM2-1 pomocí pole definicí obrazovek
<i>fbSoilMoisture</i>	Měření vlhkosti půdy pro C-HM-0308M, C-HM-1113 a C-HM-1121

6.1 Funkční blok fbC_RC_0002R

Knihovna : *CFoxLib*

Funkční blok `fbC_RC_0002R` slouží pro obsluhu modulu C-RC-0002R. Tento blok implementuje tyto funkce:

- zobrazení hodnoty (typicky teplota v prostoru)
- editace teploty (typicky žádaná teplota)
- přepínání režimu (typicky útlum / komfort)

Zobrazená čísla jsou v rozsahu -9,9 až 99,9 s jedním desetinným místem a -99 až 999 bez desetinných míst. Standardně je zobrazena hodnota přivedená na vstup `Temp`. Po zmačknutí jednoho z tlačítek modulu je zobrazena editovatelná hodnota ze vstupu `TempReq`. Tuto hodnotu lze měnit pomocí tlačítek UP a DOWN. Změna hodnoty má v rozsahu -9,5 až 99,5 krok 0,5, pro hodnoty nižší než -9,5 a vyšší než 99,5 je editační krok 1. První stisk tlačítka hodnotu pouze zobrazí, až následující stisk nebo podržení tlačítka déle jak půl sekundy způsobí změnu hodnoty.








Editovaná hodnota je po posledním stisku tlačítka zobrazována pět sekund. Poslední dvě sekundy hodnota bliká.

Tlačítko MODE alternuje proměnnou na vstupu `Mode`. Stav proměnné na vstupu `Mode` je indikován na modulu LED diodou se symbolem měsíčku. LED dioda je rozsvícena pokud je hodnota proměnné TRUE.

Zápis do proměnných na vstupech `TempReq` a `Mode` probíhá jen při změně, což umožňuje ovládní z více míst.

Spárování funkčního bloku s konkrétním modulem je realizováno přes vstupy `Btn` a `Disp`. Na vstupu `Btn` se předává struktura `BTN` obsahující stavy tlačítek a na vstupu `Disp` struktura `DISP` řídicí zobrazení. Při volání v jazyku ST je potřeba při předávání těchto struktur použít funkci `void`, jinak bude hlášen nesoulad datového typu.

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>Temp</i>	REAL	Zobrazovaná teplota (typicky teplota v prostoru)
VAR_IN_OUT			
	<i>Btn</i>	TLIB_CRC_BTN	Struktura modulu TCIB_CRC_BTN
	<i>Disp</i>	TLIB_CRC_DISP	Struktura modulu TCIB_CRC_DISP
	<i>TempReq</i>	REAL	Zobrazená hodnota editovaná tlačítka UP a DOWN (typicky žádaná teplota)
	<i>Mode</i>	BOOL	Režim
VAR_OUTPUT			
	<i>Edit</i>	BOOL	TRUE znamená, že je zobrazena editovatelná hodnota
	<i>Err</i>	BOOL	TRUE znamená, že zobrazované číslo je mimo rozsah

Příklad programu s funkčním blokem *fbC_RC_0002R*. Proměnné *C_RC_0002R_IN* a *C_RC_0002R_OUT* jsou uživatelské názvy vstupní a výstupní zóny modulu C-RC-0002R.

```

VAR_GLOBAL RETAIN
  ReqTemp : REAL := 22.0;
  Mode    : BOOL;
END_VAR

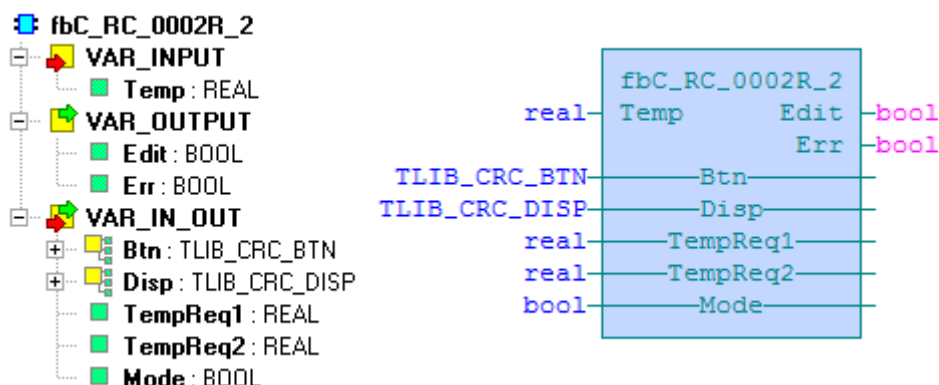
PROGRAM prgC_RC_Example
  VAR
    C_RC_1 : fbC_RC_0002R;
  END_VAR

  C_RC_1(Temp := C_RC_0002R_IN.iTHERM,
         Btn := void(C_RC_0002R_IN.BTN),
         Disp := void(C_RC_0002R_OUT.DISP),
         TempReq := ReqTemp,
         Mode := Mode);
END_PROGRAM

```

6.2 Funkční blok fbC_RC_0002R_2

Knihovna : CFoxLib



Funkční blok *fbC_RC_0002R_2* je variantou bloku *fbC_RC_0002R*. Blok se liší možností editovat pro každý režim vlastní žádanou teplotu. Hodnota na vstupu *TempReq1* je editována v případě, když je proměnná *Mode* nastavena na FALSE. Hodnota *TempReq2* pokud je *Mode* TRUE.

Typické použití může být editace žádané teploty pro běžný provoz a útlum.

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>Temp</i>	REAL	Zobrazovaná teplota (typicky teplota v prostoru)
VAR_IN_OUT			
	<i>Btn</i>	TLIB_CRC_BTN	Struktura modulu TCIB_CRC_BTN
	<i>Disp</i>	TLIB_CRC_DISP	Struktura modulu TCIB_CRC_DISP
	<i>TempReq1</i>	REAL	Zobrazená hodnota editovaná tlačítka UP a DOWN v případě <i>Mode</i> rovno FALSE (typicky žádaná teplota pro běžný provoz)
	<i>TempReq2</i>	REAL	Zobrazená hodnota editovaná tlačítka UP a DOWN v případě <i>Mode</i> rovno TRUE (typicky žádaná teplota pro útlum)
	<i>Mode</i>	BOOL	Režim
VAR_OUTPUT			
	<i>Edit</i>	BOOL	TRUE znamená, že je zobrazena editovatelná hodnota
	<i>Err</i>	BOOL	TRUE znamená, že zobrazované číslo je mimo rozsah

Příklad programu s funkčním blokem *fbC_RC_0002R_2*:

V příkladu jsou použity bloky z knihovny RegoLib. Příklad realizuje jednoduché řízení topení s týdenním programem *Prog2*. Výstup týdenního programu je negovaně promítán do režimu indikovaného modulem C-RC-0002R.

Algoritmus umožňuje změnit režim pomocí tlačítka MODE. Tento vnucený režim trvá buď jednu hodinu (časovač *ManTimer*) nebo do první změny režimu dle týdenního programu.

Topení je ovládáno blokem *Heating* realizující spínání s hysterezí půl stupně.

Proměnné *C_RC_0002R_IN* a *C_RC_0002R_OUT* jsou uživatelské názvy vstupní a výstupní zóny modulu C-RC-0002R.

```

VAR_GLOBAL RETAIN
WeekProg : TimeProg2 Week := (
  Mon := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h ),
  Tue := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h ),
  Wed := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h ),
  Thu := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h ),
  Fri := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h ),
  Sat := ( T_ON1 := T#8h, T_OFF1 := T#22h,
           T_ON2 := T#0s, T_OFF2 := T#0s ),
  Sun := ( T_ON1 := T#8h, T_OFF1 := T#22h,
           T_ON2 := T#0s, T_OFF2 := T#0s ));

TempReqLo : REAL := 21.0;
TempReqHi : REAL := 23.0;
END_VAR

PROGRAM prgC_RC_2_Example
VAR
  Prog2      : TProg2;
  Heating    : Hyst1;
  C_RC_2     : fbC_RC_0002R_2;
  lastMode   : BOOL := 1;
  crcMode    : BOOL;
  ManTimer   : TON := ( PT := T#1h );
END_VAR

Prog2(TPg := WeekProg);

IF Prog2.Out <> lastMode THEN
  lastMode := Prog2.Out;
  crcMode := NOT lastMode;
END_IF;

ManTimer(IN := crcMode = lastMode);

IF ManTimer.Q THEN
  crcMode := NOT lastMode;
END_IF;

C_RC_2(Temp := C_RC_0002R_IN.iTHERM,
       Btn := void(C_RC_0002R_IN.BTN),
       Disp := void(C_RC_0002R_OUT.DISP),
       TempReq1 := TempReqHi,
       TempReq2 := TempReqLo,
       Mode := crcMode);

```

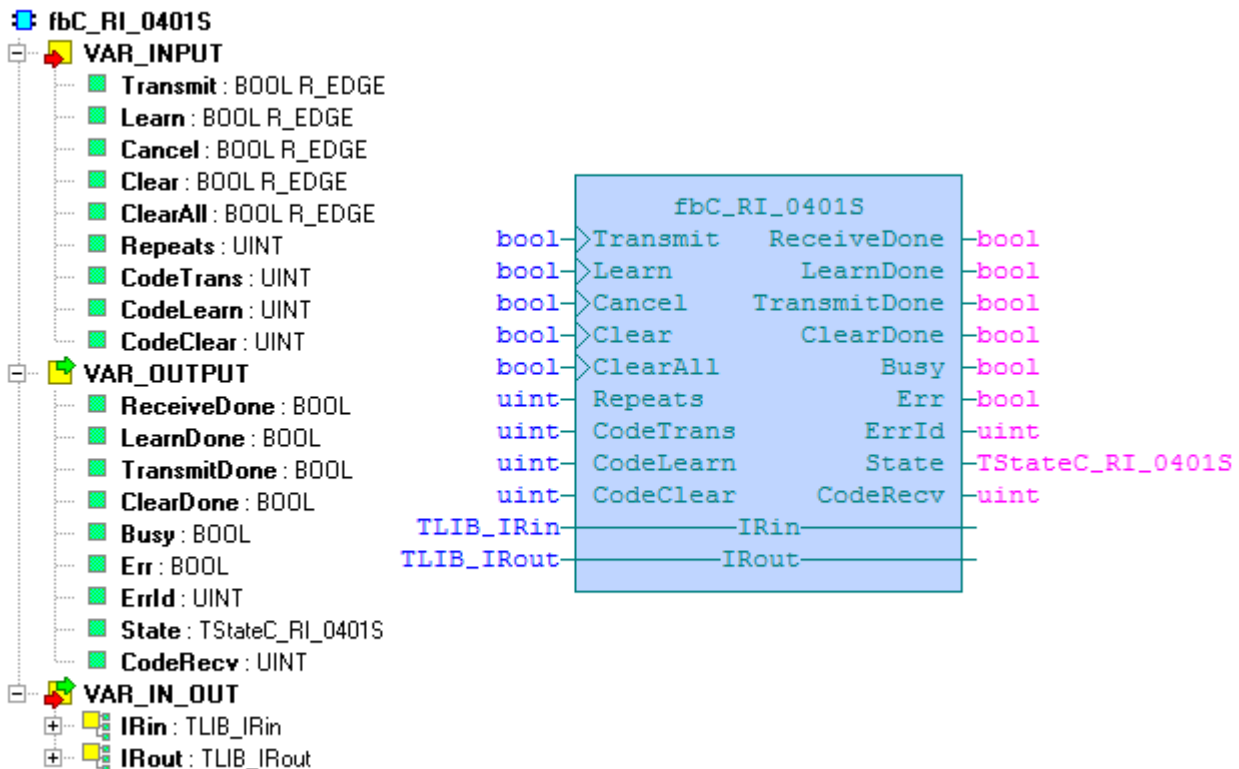
```
IF crcMode THEN
  Heating.IN := TempReqLo;
ELSE
  Heating.IN := TempReqHi;
END_IF;

Heating(PV := C_RC_0002R_IN.iTHERM, HV := 0.5);

END_PROGRAM
```

6.3 Funkční blok fbC_RI_0401S

Knihovna : CFoxLib



Funkční blok `fbC_RI_0401S` slouží k posílání, příjem a učení IR kódů pomocí modulu C-RI-0401S.

Spárování funkčního bloku s konkrétním modulem je realizováno přes vstupy `IRin` a `IRout`. Na vstupu `IRin` se předává vstupní struktura `IRin` a na vstupu `IRout` výstupní struktura modulu `IRout`. Při volání v jazyku ST je potřeba při předávání těchto struktur použít funkci `void`, jinak bude hlášen nesoulad datového typu.

Příjem:

Ve výchozím stavu blok převádí modul do režimu příjmu. Připravenost k příjmu IR kódu je indikována hodnotou `FALSE` na výstupu `Busy`. Během příjmu kódu je výstup `Busy` nastaven na `TRUE`. Úspěšné přijetí kódu je indikováno nastavením výstupu `ReceiveDone` na jeden cyklus na `TRUE`. Pro úspěšný příjem kódu je nutné, aby se tento kód modul nejdříve naučil (viz níže). Příslušný index na který byl přiřazen kódu při učení je při příjmu indikován na výstupu `CodeRecv`.

Učení:

Režim učení se iniciuje nastavením vstupu `Learn` na `TRUE`. Úspěšný přechod do režimu učení je indikováno nastavením výstupu `Busy` na `TRUE` a `State` na `sC_RI_0401S_Learning`. Při přechodu do režimu učení musí být na vstupu `CodeLearn` nenulové číslo menší než 65536 určující index kódu, který bude použit při příjmu a vysílání. Úspěšné naučení nového kódu je indikováno nastavením výstupu `LearnDone` na jeden cyklus na `TRUE`.

Vysílání:

Naučené IR kódy lze pomocí modulu vyslat. Vysílání se zahájí nastavením vstupu `Transmit` na `TRUE`. Při zahájení vysílání musí být na vstupu `CodeTrans` index kódu zvolený při učení. Některá zařízení vyžadují vícenásobné vyslání kódu, aby byl přijat. Počet opakování vysílaného kódu lze nastavit na vstupu `Repeats`. Pokud je `Repeats` roven nule IR kód je vyslán jednou, pokud je roven jedné je vyslán dvakrát, atd.

Během vysílání je výstup *Busy* nastaven na TRUE a výstup *State* nabývá hodnot *sC_RI_0401S_Trasmiting* a *sC_RI_0401S_GettingReady*. Úspěšné vyslání je indikováno nastavením výstupu *TransmitDone* na TRUE po dobu jednoho cyklu.

Mazání:

Naučené IR kódy lze z modulu vymazat. Mazání je možné buď selektivně, zadáním indexu k vymazání na vstupu *CodeClear* a nastavením *Clear* na TRUE, nebo globálně nastavením *ClearAll* na TRUE. Během mazání je výstup *Busy* nastaven na TRUE a výstup *State* nabývá hodnot *sC_RI_0401S_Clearing* a *sC_RI_0401S_GettingReady*. Úspěšné vymazání je indikováno nastavením výstupu *ClearDone* na TRUE po dobu jednoho cyklu.




















Pokud dojde během práce s IR kódy k chybě je nastaven výstup *Err* na TRUE a výstup *ErrId* na nenulovou hodnotu. Význam hodnot, kterých může výstup *ErrId* nabýt je popsán v následující tabulce:


<i>ErrId</i>	Význam
1	Chyba při příjmu. Přijatý kód neodpovídá žádnému naučenému kódu.
2	Chyba učení. Nebyl nastaven platný index pro kód k naučení nebo modul nebyl schopen zpracovat IR kód, který se měl naučit.
3	Chyba mazání. Nebylo zadán platný index pro mazání nebo se nepodařilo kód smazat.
4	Požadovaný režim není možné nastavit. Tato chyba může nastat v případě, že kromě aktuálního bloku zasahuje do vstupní struktury modulu uživatel nebo jiný blok.
5	Během 30 sekund od započetí učení nebyl přijat platný IR kód.

Během provozu blok prochází následujícími stavy:

	<i>State</i>	Význam
0	<i>sC_RI_0401S_Receiving</i>	Výchozí stav, modul je připraven k příjmu a může být převeden do jakéhokoli jiného režimu.
1	<i>sC_RI_0401S_Learning</i>	Učící mód modul čeká po dobu 30 sekund na nový IR kód k naučení.
2	<i>sC_RI_0401S_Trasmiting</i>	Modul vysílá IR kód.
3	<i>sC_RI_0401S_Clearing</i>	Modul maže jeden nebo více naučených IR kódů.
4	<i>sC_RI_0401S_Error</i>	Nastala chyba.
5	<i>sC_RI_0401S_GettingReady</i>	Přechod do výchozího stavu.

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>Transmit</i>	BOOL R_EDGE	Vyšle kód daný indexem <i>CodeTrans</i>
	<i>Learn</i>	BOOL R_EDGE	Zahájí mód učení pro index <i>CodeLearn</i>
	<i>Cancel</i>	BOOL R_EDGE	Ukončí jakýkoli probíhající mód
	<i>Clear</i>	BOOL R_EDGE	Vymaže naučený kód na indexu <i>CodeClear</i>
	<i>ClearAll</i>	BOOL R_EDGE	Vymaže všechny naučené kódy
	<i>Repeats</i>	UINT	Udává počet opakování kódu během vysílání
	<i>CodeTrans</i>	UINT	Index kódu k vyslání
	<i>CodeLearn</i>	UINT	Index kódu k naučení
	<i>CodeClear</i>	UINT	Index kódu k vymazání
VAR_IN_OUT			
	<i>IRin</i>	TLIB_IRin	Struktura <i>IRin</i> modulu C-RI-0401S
	<i>IRout</i>	TLIB_IRout	Struktura <i>IRout</i> modulu C-RI-0401S
VAR_OUTPUT			
	<i>ReceiveDone</i>	BOOL	Byl přijat nový kód
	<i>LearnDone</i>	BOOL	Modul se naučil nový kód
	<i>TransmitDone</i>	BOOL	Kód byl vyslán
	<i>ClearDone</i>	BOOL	Mazání bylo dokončeno
	<i>Busy</i>	BOOL	Modul je zaneprázdněn
	<i>Err</i>	BOOL	Nastala chyba
	<i>ErrId</i>	UINT	Identifikační kód chyby
	<i>State</i>	TStateC_RI_0401S	Stav obsluhy modulu

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
	<i>CodeRecv</i>	UINT	Index přijatého kódu

Příklad programu s funkčním blokem *fbC_RI_0401S*. Příklad demonstruje možnost učení, vysílání a přijímání 32 uživatelsky pojmenovaných kódů s vazbou na ovládání přes webové rozhraní. Uživatelská jména jsou uložena v zálohované proměnné *IrCommandNames*. Ovládání je realizováno přes proměnnou *IrCommand*, která obsahuje všechny proměnné potřebné pro ovládání přes webové rozhraní.

Význam položek struktury proměnné *IrCommand*.

Položka	Význam
<i>Index</i>	Index aktuálního IR kódu
<i>Next</i>	Přechod na o jedna vyšší index
<i>Prev</i>	Přechod na o jedna nižší index
<i>Learn</i>	Zahájení učení pro aktuální index
<i>Test</i>	Testování aktuálního indexu, očekává se příjem kódu naučeného na vybraný index. Je nastaveno automaticky po úspěšném ukončení učení.
<i>TestOk</i>	Příznak úspěšně dokončeného testu
<i>Auto</i>	Příznak automatického učení. Po naučení a otestování kódu se automaticky přejde na následující index
<i>Send</i>	Vyslání IR kódu aktuálního indexu
<i>Clear</i>	Vymazání naučeného IR kódu aktuálního indexu
<i>ClearAll</i>	Vymazání všech naučených IR kódů
<i>Transmit</i>	Řetězec 32 bitů ovládající vysílání kódů. Nastavením bitu 0 na hodnotu TRUE se vyšle kód s indexem 1, nastavením bitu 1 kód s indexem 2 atd. Pro ovládání bitu lze použít tečkový zápis (např. <i>IrCommand.Transmit.5</i>)
<i>Text</i>	Popis aktuálního indexu
<i>TextLast</i>	Poslední programem nastavený text aktuálního indexu. Neshoda s položkou <i>Text</i> je detekcí modifikace z webového rozhraní. Obsah položky <i>Text</i> je při promítnut do proměnné <i>IrCommandNames</i> .
<i>TextRecv</i>	Textový popis indexu posledního přijatého IR kódu.

Položky *Next*, *Prev*, *Learn*, *Test*, *Auto*, *Send*, *Clear*, *ClearAll*, *Transmit.0*, *Transmit.1*, ... *Transmit.31* proměnné *IrCommand* jsou navrženy pro použití jako řídicí proměnné dvoustavových obrázků ve funkci ovládacích tlačítek. Položka *TestOk* slouží jako indikace úspěšného textu. Položka *Text* může být zobrazena v editovatelném poli, které umožní uživatelskou změnu jména IR kódu.

```
TYPE
  TIrCommand : STRUCT
    Index      : UINT;
    Next       : BOOL;
    Prev       : BOOL;
    Learn      : BOOL;
    Test       : BOOL;
    TestOk     : BOOL;
    Auto       : BOOL;
    Send       : BOOL;
    Clear      : BOOL;
    ClearAll   : BOOL;
    Transmit   : DWORD;
    Text       : STRING[24];
    TextLast   : STRING[24];
    TextRecv   : STRING[24];
  END_STRUCT;
END_TYPE

VAR_GLOBAL CONSTANT
  IrCommandMax : UINT := 31;
END_VAR

VAR_GLOBAL RETAIN
  IrCommandNames : ARRAY [0..IrCommandMax] OF STRING [24] := [
    'Power',      'Mute',      'Volume +',  'Volume -',
    'Return',     'Start',     'Cancel',   'Up',
    'Down',       'Left',     'Right',    'Ok',
    'Stop',       'Play/Pause', 'Previous',  'Next',
    'Command 17', 'Command 18', 'Command 19', 'Command 20',
    'Command 21', 'Command 22', 'Command 23', 'Command 24',
    'Command 25', 'Command 26', 'Command 27', 'Command 28',
    'Command 29', 'Command 30', 'Command 31', 'Command 32'
  ];
END_VAR

VAR_GLOBAL
  IrCommand : TIrCommand;
END_VAR

FUNCTION DwordBitIndex : USINT
  VAR_INPUT
    In : DWORD;
  END_VAR
  VAR
    i : USINT;
  END_VAR

  WHILE In > 0 DO
    In := SHR(IN := In, N := 1);
    i := i + 1;
  END_WHILE;
  DwordBitIndex := i;
END_FUNCTION
```

```

PROGRAM prgC_RI_Example
VAR
  IrControl      : fbC_RI_0401S;
  IrCommandLearnEnd : F_TRIG;
  AutoTimeOk     : TON;
END_VAR

//changing index
IF IrCommand.Next THEN
  IF IrCommand.Index = IrCommandMax THEN
    IrCommand.Index := 0;
    IrCommand.Auto := false;
  ELSE
    IrCommand.Index := IrCommand.Index + 1;
  END_IF;
  IrCommand.Next := false;
  IrCommand.Learn := IrCommand.Auto;
  IrCommand.Test := false;
  IrCommand.TestOk := false;
END_IF;

IF IrCommand.Prev THEN
  IF IrCommand.Index = 0 THEN
    IrCommand.Index := IrCommandMax;
  ELSE
    IrCommand.Index := IrCommand.Index - 1;
  END_IF;
  IrCommand.Prev := false;
  IrCommand.Learn := false;
  IrCommand.Test := false;
  IrCommand.TestOk := false;
END_IF;

//changing names
IF IrCommand.TextLast <> IrCommand.Text THEN
  IrCommandNames[IrCommand.Index] := IrCommand.Text;
END_IF;

IrCommand.Text := IrCommandNames[IrCommand.Index];
IrCommand.TextLast := IrCommand.Text;

IF IrCommand.Send THEN
  IrControl.Transmit := true;
  IrControl.CodeTrans := IrCommand.Index+1;
END_IF;

//transmitting command
IF IrCommand.Transmit <> 0 THEN
  IrControl.Transmit := true;
  IrControl.CodeTrans := USINT_TO_UINT(
    DwordBitIndex(In := IrCommand.Transmit)
  );
END_IF;

IrControl(Clear := IrCommand.Clear,
  ClearAll := IrCommand.ClearAll,
  Learn := IrCommand.Learn,
  CodeLearn := IrCommand.Index+1,
  CodeClear := IrCommand.Index+1,
  IRin := void(C_RI_0401S_IN.IRin),
  IRout := void(C_RI_0401S_OUT.IRout));

```

```
//clearing requests
IF NOT IrControl.Busy THEN
  IrCommand.Transmit := 0;
  IrCommand.Send := 0;
  IrCommand.Clear := 0;
  IrCommand.ClearAll := 0;
END_IF;

IrCommand.Learn := IrCommand.Learn AND
  IrControl.state = sC_RI_0401S_Learning;

IrCommandLearnEnd(CLK := IrCommand.Learn);

IrControl.Cancel := IrControl.state = sC_RI_0401S_Learning
  AND IrCommandLearnEnd.Q;

IrCommand.Test := IrCommand.Test OR IrControl.LearnDone;

//identifying of received message
IF IrControl.ReceiveDone THEN
  IF IrControl.CodeRecv > 0 AND
    IrControl.CodeRecv <= IrCommandMax THEN
    IrCommand.TextRecv := IrCommandNames[IrControl.CodeRecv-1];
  ELSE
    IrCommand.TextRecv := 'unknown';
  END_IF;
  //test result
  IF IrCommand.Test THEN
    IrCommand.TestOk := IrControl.CodeLearn = IrControl.CodeRecv;
    IrCommand.Test := false;
  END_IF;
END_IF;

IrCommand.TestOk := IrCommand.TestOk AND (NOT IrCommand.Learn);

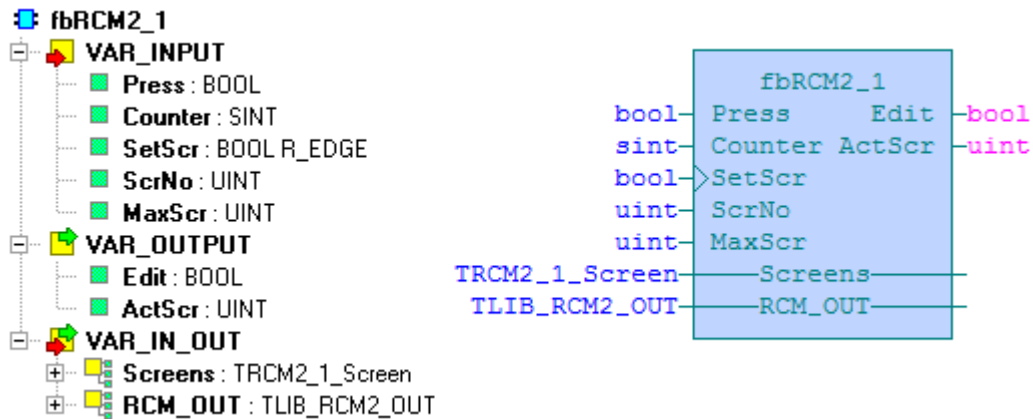
//automatic learning
AutoTimeOk(IN := IrCommand.Auto AND IrCommand.TestOk, PT := T#2s);

IF IrCommand.Auto THEN
  IF AutoTimeOk.Q THEN
    IrCommand.Next := true;
  END_IF;
END_IF;

END_PROGRAM
```

6.4 Funkční blok fbRCM2_1

Knihovna : CFoxLib



Funkční blok `fbRCM2_1` je určen pro definici a obsluhu lineárního seznamu obrazovek zobrazovaných modulem RCM2-1. Spárování funkčního bloku s konkrétním modulem je realizováno přes vstupy `Press`, `Counter` a `RCM_OUT`. Do vstupu `Press` a `Counter` je se přiřadí odpovídající proměnné ze vstupní struktury modulu jmenovitě `FLG.PRESS` a `Counter`. Výstupní struktura modulu se předává na vstupu `RCM_OUT`. Při volání v jazyku ST je potřeba při předávání této struktury použít funkci `void`, jinak bude hlášen nesoulad datového typu.

Definice obrazovek se provádí pomocí pole struktur předávané na vstupu `Screens`. Každá položka pole představuje popis jedné obrazovky. Mezi obrazovkami lze přecházet pomocí otočného ovladače modulu nebo je možné vnutit zobrazení obrazovky pomocí vstupu `SetScr`. Při nastavení `SetScr` na `TRUE` dojde k zobrazení obrazovky s indexem daným vstupem `ScrNo`. Indexy obrazovek začínají jedničkou bez ohledu na indexování pole definic. Pro přehlednost algoritmu je však vhodné toto pole indexovat od jedné. Pole může mít libovolný počet položek, který se předá bloku na vstupu `MaxScr`. **MaxScr nesmí být nikdy větší než rozměr pole!**

Výstup `Edit` indikuje probíhající editaci na aktuálně zobrazené obrazovce. Index aktuálně zobrazené obrazovky udává výstup `ActScr`.












Definice obrazovky obsahuje zobrazovanou hodnotu (typ `REAL` nebo `TIME`), zobrazené symboly, počet zobrazených desetinných míst a příznaky povolující a indikující editaci.













Podrobný výčet položek struktury popisujíc obrazovku:

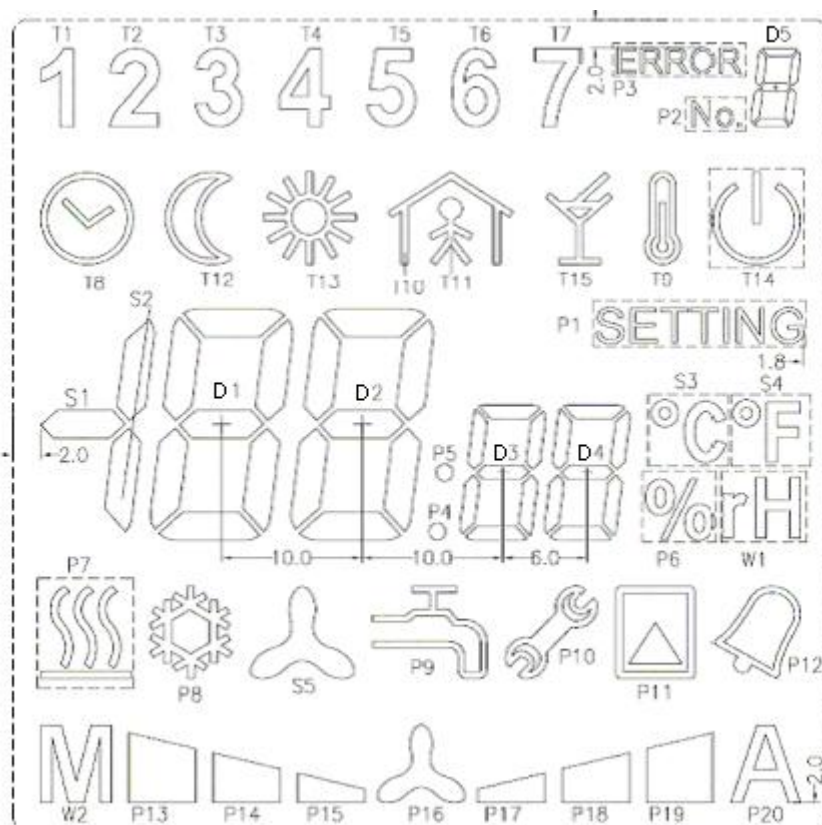
Položka	Význam
<i>RealValue</i>	Zobrazovaná hodnota REAL
<i>RealInc</i>	Inkrement při otočení kolečka o jeden krok pro REAL
<i>RealMin</i>	Minimum REAL hodnoty při editaci
<i>RealMax</i>	Maximum REAL hodnoty při editaci
<i>TimeValue</i>	Zobrazovaná hodnota TIME
<i>TimeMin</i>	Minimum TIME hodnoty při editaci
<i>TimeMax</i>	Maximum TIME hodnoty při editaci
<i>IsTime</i>	Příznak pro použití hodnoty TIME místo výchozí REAL
<i>Editing</i>	Probíhá editace
<i>EditEnd</i>	Editace dokončena a potvrzená
<i>Editable</i>	Zobrazenou hodnotu lze editovat pokud je TRUE
<i>BlinkingDots</i>	Zobrazí se blikající dvojtečka
<i>ShowTenths</i>	Zobrazí jedno desetinné místo
<i>ShowHundredths</i>	Zobrazí dvě desetinná místa
<i>Symbols</i>	Symbole které se mají na obrazovce zobrazit. Struktura odpovídá výstupní struktuře <i>ICO</i> modulu RCM2-1

Popis struktury *Symbols*

Položka	Význam
<i>ONE</i>	Zobrazí symbol T1 1
<i>TWO</i>	Zobrazí symbol T2 2
<i>THREE</i>	Zobrazí symbol T3 3
<i>FOUR</i>	Zobrazí symbol T4 4
<i>FIVE</i>	Zobrazí symbol T5 5
<i>SIX</i>	Zobrazí symbol T6 6
<i>SEVEN</i>	Zobrazí symbol T7 7
<i>ERROR</i>	Zobrazí symbol P3 ERROR
<i>No</i>	Zobrazí symbol P2 No.

<i>Clock</i>	Zobrazí symbol T8	
<i>Moon</i>	Zobrazí symbol T12	
<i>Sun</i>	Zobrazí symbol T13	
<i>House</i>	Zobrazí symbol T10	
<i>Figure</i>	Zobrazí symbol T11	
<i>Drink</i>	Zobrazí symbol T15	
<i>Thermometer</i>	Zobrazí symbol T9	
<i>Power</i>	Zobrazí symbol T14	
<i>Setting</i>	Položka je ovládána funkčním blokem v režimu editace. SETTINGS	
<i>Celsius</i>	Zobrazí symbol S3	°C
<i>Fahrenheit</i>	Zobrazí symbol S4	°F
<i>Percent</i>	Zobrazí symbol P6	%
<i>rH</i>	Zobrazí symbol W1	rH
<i>dotUp</i>	Nastavení nemá vliv, položka je ovládána funkčním blokem ▪	
<i>dotDown</i>	Nastavení nemá vliv, položka je ovládána funkčním blokem ▪	
<i>Heating</i>	Zobrazí symbol P7	
<i>Cooling</i>	Zobrazí symbol P8	
<i>Ventilation</i>	Zobrazí symbol S5	

<i>Water</i>	Zobrazí symbol P9	
<i>Spanner</i>	Zobrazí symbol P10	
<i>P11</i>	Zobrazí symbol P11	
<i>Bell</i>	Zobrazí symbol P12	
<i>Manual</i>	Zobrazí symbol W2	M
<i>LN3</i>	Zobrazí symbol P13	
<i>LN2</i>	Zobrazí symbol P14	
<i>LN1</i>	Zobrazí symbol P15	
<i>Rotation</i>	Zobrazí symbol P16	
<i>LP1</i>	Zobrazí symbol P17	
<i>LP2</i>	Zobrazí symbol P18	
<i>LP3</i>	Zobrazí symbol P19	
<i>Automatic</i>	Zobrazí symbol P20	A
<i>Minus</i>	Nastavení nemá vliv, položka je ovládána funkčním blokem	
<i>S2</i>	Nastavení nemá vliv, položka je ovládána funkčním blokem	
<i>D1</i>	Nastavení nemá vliv, položka je ovládána funkčním blokem	
<i>D2</i>	Nastavení nemá vliv, položka je ovládána funkčním blokem	
<i>D3</i>	Nastavení nemá vliv, položka je ovládána funkčním blokem	
<i>D4</i>	Nastavení nemá vliv, položka je ovládána funkčním blokem	
<i>DE</i>	Zobrazí symbol D5	
<i>DOUT</i>	Nastaví binární výstup	



Rozmístění a označení symbolů a segmentů na displeji RCM2-1

Ovládání displeje:










V běžném režimu lze pomocí tlačítka přecházet mezi nadefinovanými obrazovkami. Otočení vpravo zobrazí následující obrazovku, otočení vlevo předchozí. Procházení obrazovek se děje v kruhu, což znamená, že na poslední obrazovce se otočením po směru hodinových ručiček zobrazí první obrazovka a na první obrazovce otočením proti směru se zobrazí poslední obrazovka. Pokud má obrazovka povolenu editaci, stisknutím točítka dojde k jejímu zahájení.

Editace hodnoty je signalizována blikáním segmentu P1 (SETTINGS). V programu je během editace nastaven výstup bloku *Edit* a položka *Editing* v popisu aktuálně zobrazené obrazovky na TRUE. Během editace není položka *RealValue* (respektive *TimeValue*) modifikována dokud nedojde k potvrzení editace zmačknutím točítka. Při potvrzení editace se nastaví na jeden cyklus PLC položka *EditEnd*. Režim editace je v případě nečinnosti ukončen automaticky po deseti sekundách s tím, že nedojde ke změně *RealValue* (respektive *TimeValue*) ani k nastavení *EditEnd*.

Při editaci hodnoty typu REAL bliká počet editovaných cifer daný velikostí inkrementu *RealInc*. Otočení o jednu aretaci po směru hodinových ručiček zvýší hodnotu o *RealInc*, otočení proti směru ji naopak sníží.

Editace času probíhá ve dvou fázích. Při zahájení editace bliká údaj hodin a otočení o jednu aretaci zvýší nebo sníží o jednu hodinu. Po potvrzení hodinového údaje stisknutím točítka se rozblikají a jdou editovat číslice udávající počet minut. Otočením točítka se v této fázi mění údaj po jedné minutě. Další potvrzení stiskem točítka upravenou hodnotu času uloží

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>Press</i>	BOOL	Proměnná <i>PRESS</i> ze vstupní struktury RCM2-1
	<i>Counter</i>	SINT	Proměnná <i>Counter</i> ze vstupní struktury RCM2-1
	<i>SetScr</i>	BOOL R_EDGE	Změní aktuální zobrazovanou obrazovku
	<i>ScrNo</i>	UINT	Číslo obrazovky, na kterou se má přejít při nastavení <i>SetScr</i>
	<i>MaxScr</i>	UINT	Počet obrazovek (nesmí být větší než rozměr pole definicí obrazovek)
VAR_IN_OUT			
	<i>Screens</i>	TRCM2_1_Screen	Struktura modulu TCIB_CRC_BTN
	<i>RCM_OUT</i>	TLIB_RCM2_OUT	Struktura modulu TCIB_CRC_DISP
VAR_OUTPUT			
	<i>Edit</i>	BOOL	TRUE znamená, že aktuálně zobrazená hodnota je editována
	<i>ActScreen</i>	UINT	Aktuálně zobrazená obrazovka

Následující příklad programu s funkčním blokem *fbRCM2_1* definuje 5 obrazovek umožňující zobrazení aktuálního času, pokojové teploty, editaci dvou žádaných teplot a nastavení času alarmu. V příkladu je použit časový program *TProg2* z knihovny *RegoLib*. Pro větší přehlednost je každé z obrazovek v poli definic *RCM_Screens* definováno jméno s vlastní inicializací.

Obrazovka *ClockScr* zobrazuje aktuální čas, den v týdnu a režim časového programu.

Obrazovka *RoomTempScr* zobrazuje teplotu z externího čidla modulu, den v týdnu a režim časového programu.

Obrazovka *ReqTempHiScr* umožňuje editaci žádané teploty *TempReqHi*.

Obrazovka *ReqTempLoScr* umožňuje editaci žádané teploty *TempReqLo*.

Obrazovka *AlarmSetScr* umožňuje editaci časového údaje *TimeAlarm*.

Časovač *RCM2_1_BackToFirst* zajišťuje zobrazení první obrazovky pokud po dobu jedné minuty nedojde k editaci nebo změně obrazovky.

Pro synchronizaci editovaných hodnot jsou využity funkce *RCM2_1_SyncVarReal* a *RCM2_1_SyncVarTime*.

V příkladu je dále demonstrována možnost dynamického zobrazení symbolů. U obrazovek *ClockScr* a *RoomTempScr* je indikace stavu časového programu řešena zápisem do pole definic obrazovek. Zobrazení dne v týdnu je řešeno přímým zápisem do zóny displeje.

```

VAR GLOBAL RETAIN
WeekProg : TimeProg2_Week := (
  Mon := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h),
  Tue := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h),
  Wed := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h),
  Thu := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h),
  Fri := ( T_ON1 := T#6h, T_OFF1 := T#7h,
           T_ON2 := T#16h, T_OFF2 := T#22h),
  Sat := ( T_ON1 := T#8h, T_OFF1 := T#22h,
           T_ON2 := T#0s, T_OFF2 := T#0s),
  Sun := ( T_ON1 := T#8h, T_OFF1 := T#22h,
           T_ON2 := T#0s, T_OFF2 := T#0s));
TempReqLo : REAL := 21.0;
TempReqHi : REAL := 23.0;
TimeAlarm : TIME := T#7h;
END_VAR

VAR GLOBAL
RCM_Screens : ARRAY [1..5] OF TRCM2_1_Screen;
ClockScr AT RCM_Screens[1] : TRCM2_1_Screen :=
  (IsTime := true, BlinkingDots := true,
   Symbols := ( Clock := true));
RoomTempScr AT RCM_Screens[2] : TRCM2_1_Screen :=
  (ShowTenths := true,
   Symbols := ( Thermometer := true, Celsius := true));
ReqTempHiScr AT RCM_Screens[3] : TRCM2_1_Screen :=
  (RealInc := 0.5, RealMin := 15.0, RealMax := 30.0,
   ShowTenths := true, Editable := true,
   Symbols := ( Thermometer := true, Sun := true,
               Spanner := true, Celsius := true));
ReqTempLoScr AT RCM_Screens[4] : TRCM2_1_Screen :=
  (RealInc := 0.5, RealMin := 15.0, RealMax := 30.0,
   ShowTenths := true, Editable := true,
   Symbols := ( Thermometer := true, Moon := true,
               Spanner := true, Celsius := true));
AlarmSetScr AT RCM_Screens[5] : TRCM2_1_Screen :=
  (TimeMin := T#0s, TimeMax := T#23h59m,
   IsTime := true, Editable := true,
   Symbols := ( Clock := true, Spanner := true,
               Bell := true));
END_VAR

PROGRAM prgRCM2_Example
VAR
  Prog2 : TProg2;
  RCM2_1 : fbRCM2_1;
  RCM2_1_BackToFirst : TON := (PT := T#1m);
  LastScreen : UINT;
END_VAR

Prog2 (TPg := WeekProg);

ClockScr.TimeValue := GetTime();
ClockScr.Symbols.Moon := NOT Prog2.Out;
ClockScr.Symbols.Sun := Prog2.Out;

RoomTempScr.RealValue := RCM2_1.IN.eTHERM;
RoomTempScr.Symbols.Moon := NOT Prog2.Out;
RoomTempScr.Symbols.Sun := Prog2.Out;

```

```
RCM2_1_SyncVarReal(Val := TempReqHi, Screen := ReqTempHiScr);
RCM2_1_SyncVarReal(Val := TempReqLo, Screen := ReqTempLoScr);
RCM2_1_SyncVarTime(Val := TimeAlarm, Screen := AlarmSetScr);
RCM2_1(Press := RCM2_1_IN.FLG.PRESS,
      Counter := RCM2_1_IN.Counter,
      MaxScr := 5,
      Screens := RCM_Screens[1],
      RCM_OUT := void(RCM2_1_OUT));

IF NOT RCM_Screens[RCM2_1.ActScr].Editable THEN
  RCM2_1_OUT.ICO.ONE := System_S.COUNTER_DAYS_OF_WEEK = 1;
  RCM2_1_OUT.ICO.TWO := System_S.COUNTER_DAYS_OF_WEEK = 2;
  RCM2_1_OUT.ICO.THREE := System_S.COUNTER_DAYS_OF_WEEK = 3;
  RCM2_1_OUT.ICO.FOUR := System_S.COUNTER_DAYS_OF_WEEK = 4;
  RCM2_1_OUT.ICO.FIVE := System_S.COUNTER_DAYS_OF_WEEK = 5;
  RCM2_1_OUT.ICO.SIX := System_S.COUNTER_DAYS_OF_WEEK = 6;
  RCM2_1_OUT.ICO.SEVEN := System_S.COUNTER_DAYS_OF_WEEK = 7;
END_IF;

RCM2_1_BackToFirst(IN := NOT RCM2_1.Edit AND
                  RCM2_1.ActScr = LastScreen AND
                  RCM2_1.ActScr <> 1);

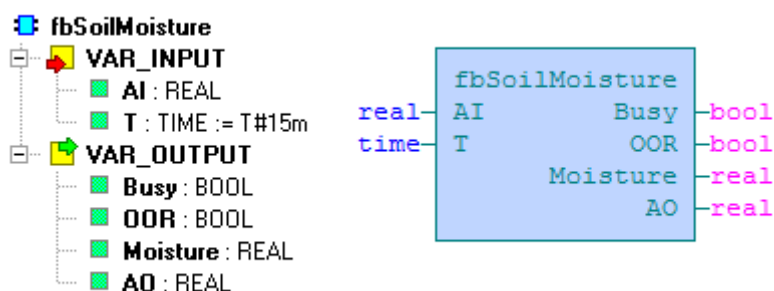
LastScreen := RCM2_1.ActScr;

IF RCM2_1_BackToFirst.Q THEN
  RCM2_1.SetScr := 1;
  RCM2_1.ScrNo := 1;
END_IF;

END_PROGRAM
```

6.5 Funkční blok fbSoilMoisture

Knihovna : CFoxLib



Funkční blok *fbSoilMoisture* slouží pro řízení měření objemové vlhkosti půdy pro moduly C-HM-0308M, C-HM-1113 a C-HM-1121. Blok zajišťuje měření v doporučeném intervalu 15 minut. Interval měření lze ovlivnit nastavením vstupu *T*. Probíhající měření je signalizováno nastavením výstupu *Busy*. Výsledná hodnota objemové vlhkosti je předávána na výstupu *Moisture* v rozsahu 5 až 50%. V případě přetečení nebo podtečení rozsahu je výstup *OOR* nastaven na hodnotu TRUE a na výstupu *Moisture* ponechána okrajová hodnota.

Spojení bloku s příslušným modulem je realizováno přes vstup *AI*, kde se předává hodnota analogového vstupu s rozsahem 0 až 2 V a výstupem *AO*, kde musí být přiřazen analogový výstup použitý k buzení čidla vlhkosti.

Zapojení a typ čidla viz Příručka projektanta systémů CFox a RFox (TXV 004 16) kapitola 12.4.2.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>AI</i>	REAL	Analogový vstup modulu C-HM s rozsahem 0 až 2 V
	<i>T</i>	TIME	Perioda měření
VAR_OUTPUT			
	<i>Busy</i>	BOOL	TRUE znamená, že probíhá měření
	<i>OOR</i>	BOOL	TRUE znamená, že znamená že měřená hodnota je mimo meze
	<i>Moisture</i>	REAL	Objemová vlhkost půdy 5 až 50%
	<i>AO</i>	REAL	Hodnota pro analogový výstup modulu C-HM

Příklad programu s funkčním blokem *fbSoilMoisture*:

```
VAR_GLOBAL
  gSoilMoisture : REAL;
END_VAR

PROGRAM prgSoilMoisture
  VAR
    SoilMoisture : fbSoilMoisture;
  END_VAR

  SoilMoisture(AI := C_HM_0308M_IN.AI.AI1, AO => C_HM_0308M_OUT.AO.AO1);

  gSoilMoisture := SoilMoisture.Moisture;
END_PROGRAM
```



teco

Objednávky a informace:

Teco a. s. Havlíčkova 260, 280 58 Kolín 4, tel. 321 737 611, fax 321 737 633

TXV 003 71.01

Výrobce si vyhrazuje právo na změny dokumentace. Poslední aktuální vydání je k dispozici na internetu www.tecomat.cz