



Scalable PLC AC500

# AC500

## SD Memory Card

## Basic module

# SD Memory Card Contents



- General
- The file structure
- Store / load the project to / from the card
- Source code download and upload
- READ / WRITE data
- READ / WRITE retentive data
- Firmware update for PM5xx

# SD Memory Card Coming up



- **General**
- The file structure
- Store / load the project to / from the card
- Source code download and upload
- READ / WRITE data
- READ / WRITE retentive data
- Firmware update for PM5xx

# SD Memory Card General



- Used by AC500 like an auxiliary disk drive
- For data transfer between a PC with SD card reader and AC500
- For loading and saving of
  - Source code of user program
  - Retentive variables
  - User data
  - User program (boot project)
- For firmware update of
  - AC500 CPU and / or boot code update
  - Display controller
  - Communication modules
- Option, not necessary to run user program

# SD Memory Card Coming up



- General
- **The file structure**
- Store / load the project to / from the card
- Source code download and upload
- READ / WRITE data
- READ / WRITE retentive data
- Firmware update for PM5xx

# SD Memory Card

## The File Structure Created by PLC

Name	Size	Type
FIRMWARE		File Folder
SDCARD.INI	1 KB	Configuration Settings
USERDATA		File Folder

- SD card is accessed like a PC floppy disk drive.
- File system type is FAT (MS DOS format).
- File names are to be of max. 8 characters.

### File structure creation

- Plug the formatted SD card into the slot of an already powered PLC.
- The RUN LED is blinking fast.
- After installation is done, the blinking is stopped.
- The structure can be also manually created by use of PC.

**Note:** As of version 2.0.x the subfolders of folders FIRMWARE and USERDATA are created only for the matching CPU type.

# SD Memory Card

## The File Structure as of V1.2 (1)

- The file structure **as of V1.2** is created by use of a self-extracting zip file.
- A zip file containing FW and/or boot code files for the CPUs and/or FW for communication modules can be downloaded from the ABB website:
  - Go to [www.abb.com/plc](http://www.abb.com/plc).
  - Follow the link: **Firmware Updates**.
- Select the appropriate link e.g.
  - ⊕ [FW5xx-UPDT, FW update files V1.3.2 / V1.3.4 for AC500 and AC500-eCo CPUs and Couplers: PM55x, PM56x and PM57x, PM58x, PM59x with/without ETH/ARCNET, CM572, CM574, CM575, CM577, CM578](#)  
English, French, German - 22,08MB

### More Info Links

[CAD Drawing Library](#)  
[PLC's Literature Library](#)  
[I/O's Literature Library](#)  
[Firmware Updates](#)  
[PS501 Updates](#)  
[Documentation Updates](#)

# SD Memory Card

## The File Structure as of V1.2 (2)

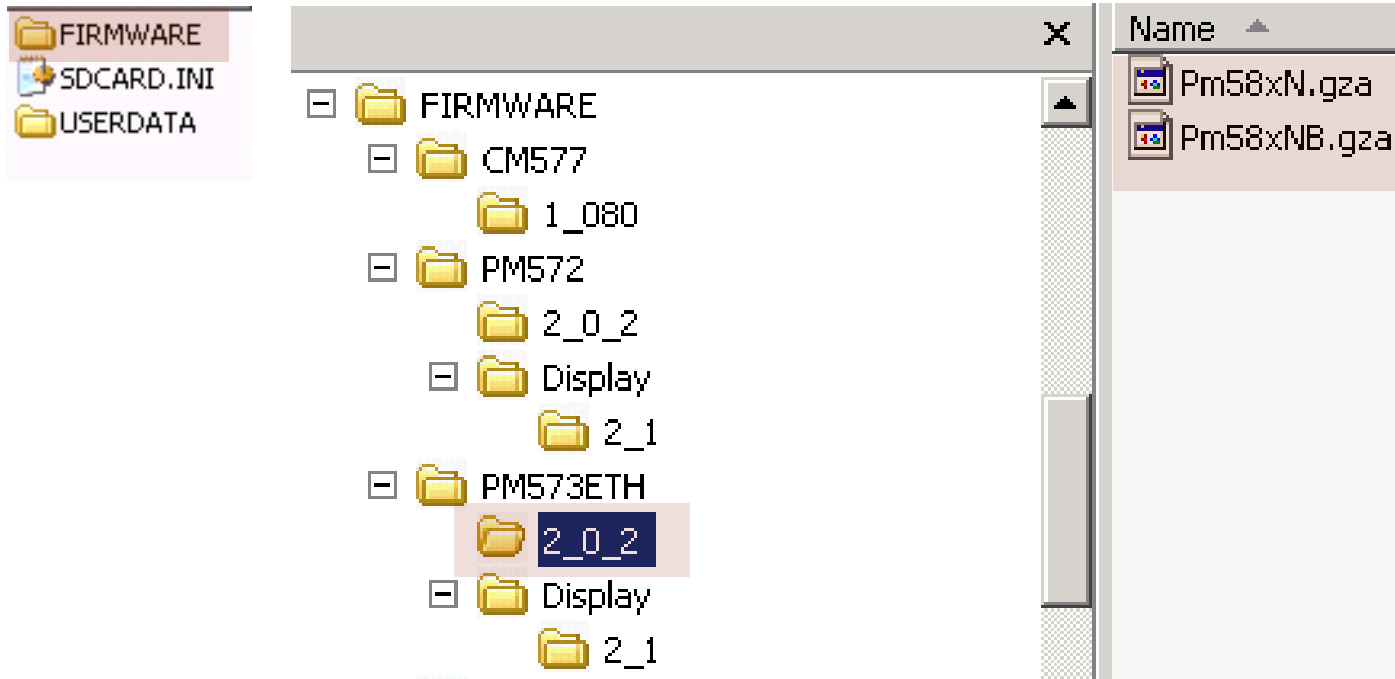
- Download the file to a desired directory.
- Open the zip file and extract the contents.
- The zip file contains:
  - A self-extracting file to create the SD card directory structure and data files “**SDCard\_create\*.exe**”.
  - Update instructions in several languages: English, French and German.
- Click on the “**SD card\_create\*.exe**” file to start the procedure.
- Select the SD card reader/writer root directory to extract the file.

**Note:** The .exe file provides structure containing folders FIRMWARE and USERDATA and subfolders for all types of components to be updated (e.g. all ETH components). The missing subfolders will be created by CPU after inserting the SD card into its slot.



# SD Memory Card

## Folder Firmware: CPUs

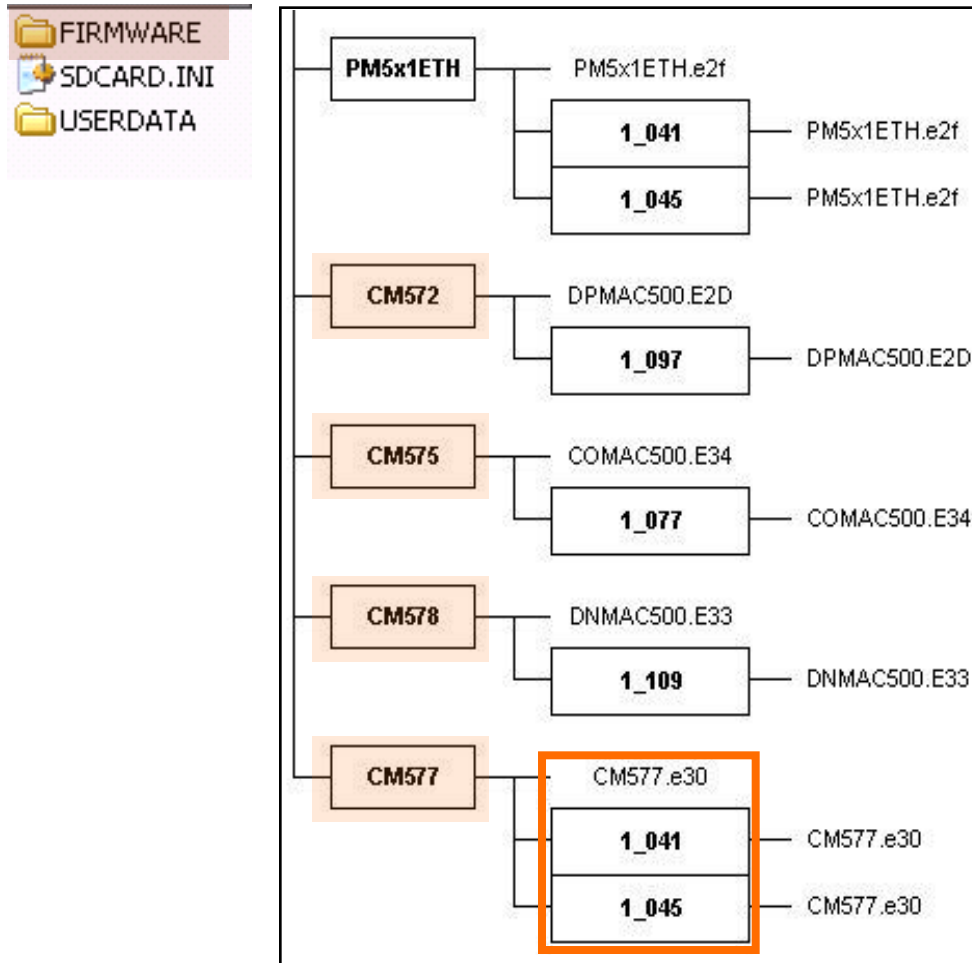


Folder created by .exe file. Each module type has its own directory.

Management of several boot code and firmware versions on one SD card is possible.

# SD Memory Card

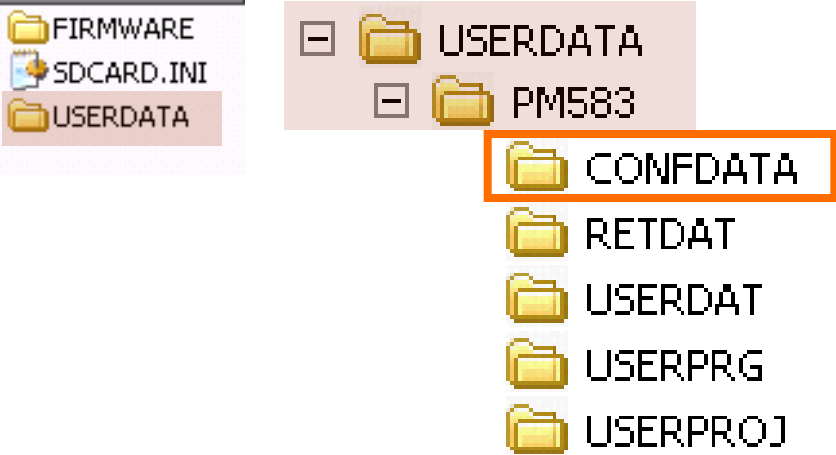
## Folder Firmware: Communication Modules



- Folder created by .exe file.
  - Management of several firmware versions for one module on one SD card is possible.
  - Loading the module firmware from the SD card:
    - by specific settings in the file SDCARD.INI
- or
- by using the PLC browser command.

# SD Memory Card

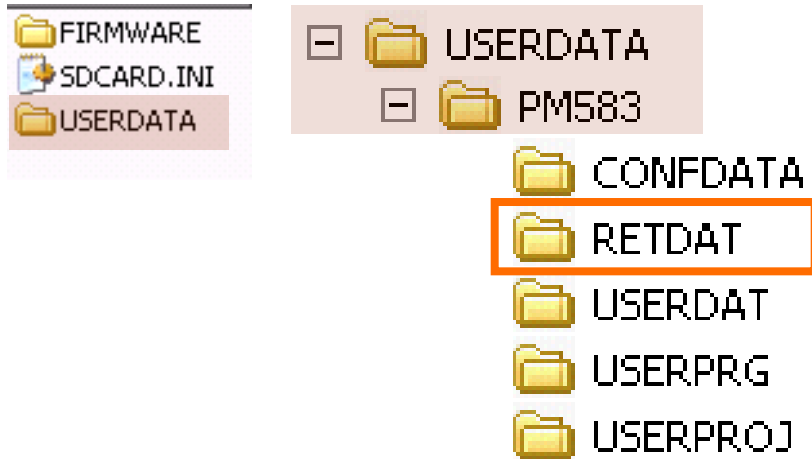
## Folder USERDATA\PM5xx\CONFDATA



The folder CONFDATA is reserved for use in the future.

# SD Memory Card

## Folder USERDATA\PM5xx\RETDAT

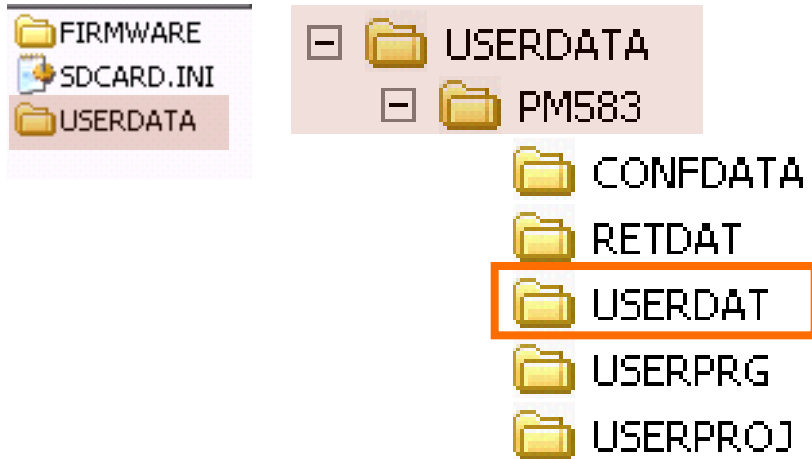


Saving / restoring data to / from the SD card via the user program and / or the PLC browser:

- RETAIN data (%M area excluded)
- PERSISTENT area (%R area) defined as buffered.

# SD Memory Card

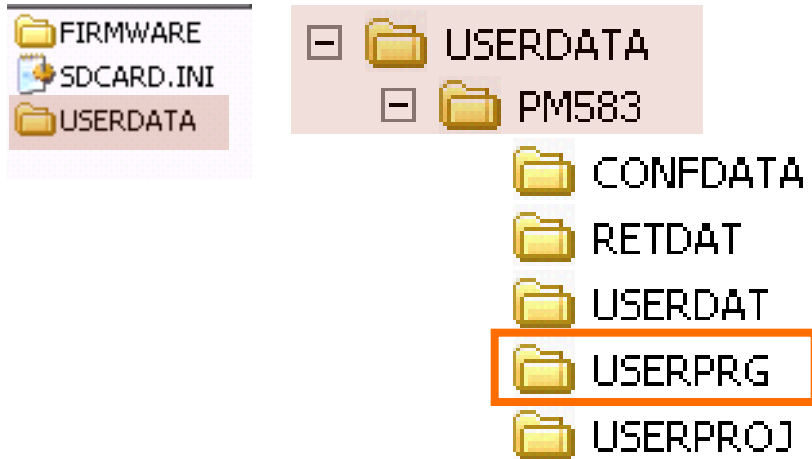
## Folder USERDATA\PM5xx\USERDAT



- A maximum of 100 files (**USRDAT00.DAT...USRDAT99.DAT**) created by user program can be stored in one directory.
- Each data file **USRDATxx.DAT** can be divided into individual sectors, if necessary.

# SD Memory Card

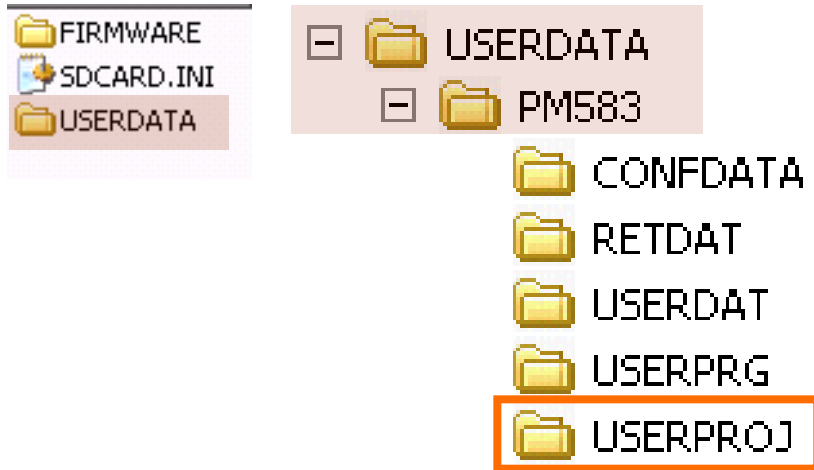
## Folder USERDATA\PM5xx\USERPRG



- The boot project files (DEFAULT.PRG and DEFAULT.CHK) and configuration files for communication modules are packed in the compressed file **BOOT.ZIP**.
- Duplication of the machine code and communication modules' settings of the PLC (duplication for PLC devices by use of SD card only).

# SD Memory Card

## Folder USERDATA\PM5xx\USERPROJ



- The project sources and all parts of a project are packed in the compressed file "**SOURCE.DAT**".
- The file allows the backup of PS501 project (project duplication for programming device).

# SD Memory Card Coming up

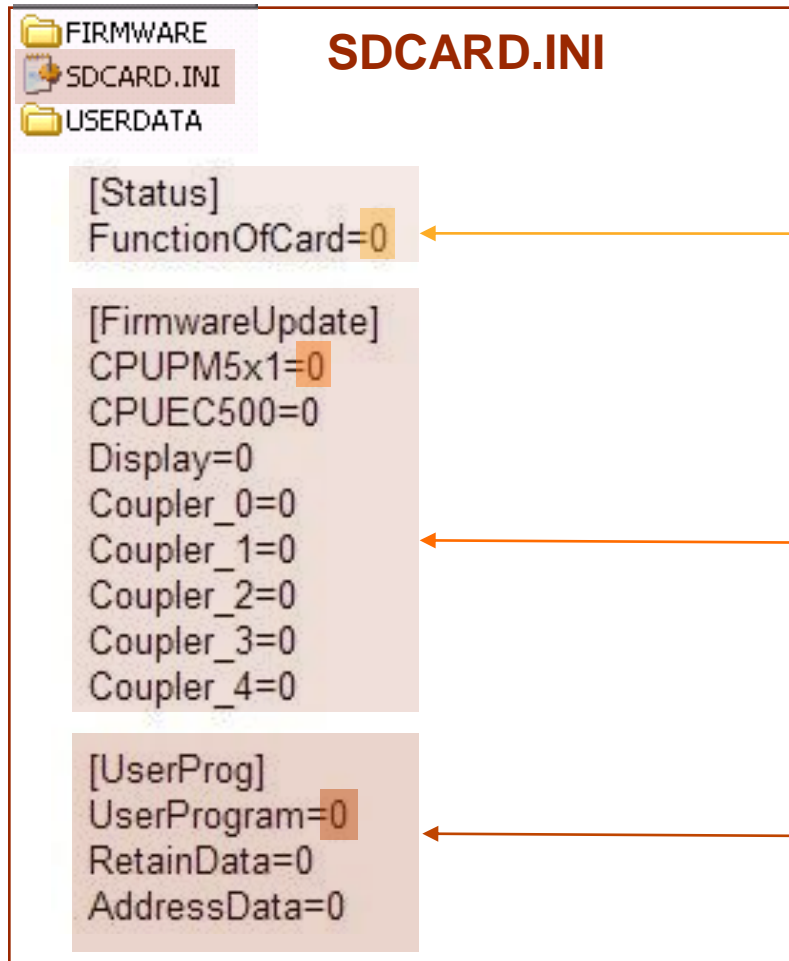


- The file structure
- Store / load the project to / from the card
- Source code download and upload
- READ / WRITE data
- READ / WRITE retentive data
- Firmware update for PM5xx



# SD Memory Card

## The Command File SDCARD.INI



The image shows a file explorer view of the SDCARD.INI file. The file is located in the root directory of the SD card, alongside folders for FIRMWARE and USERDATA. The file content is organized into three sections: [Status], [FirmwareUpdate], and [UserProg]. Each section contains several configuration parameters, with the values 0, 1, 2, and 3 highlighted in orange boxes. Arrows point from these highlighted values to the corresponding legend boxes on the right.

```
[Status]
FunctionOfCard=0

[FirmwareUpdate]
CPUPM5x1=0
CPU500=0
Display=0
Coupler_0=0
Coupler_1=0
Coupler_2=0
Coupler_3=0
Coupler_4=0

[UserProg]
UserProgram=0
RetainData=0
AddressData=0
```

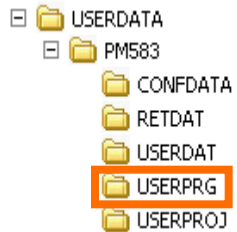
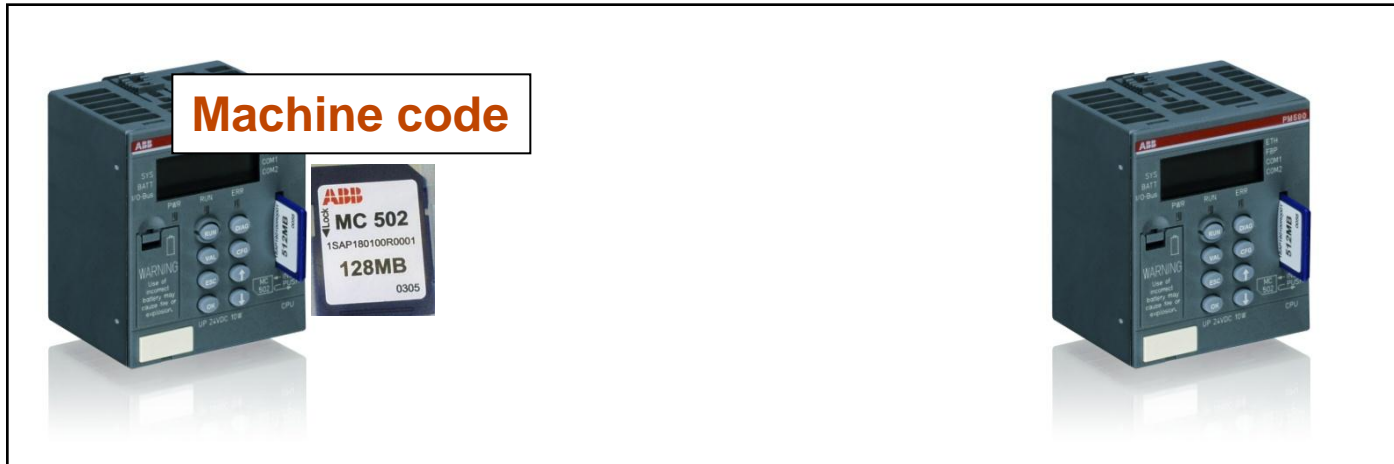
- 0: no function
- 1: load user program
- 2: firmware update as defined (see below)
- 3: update firmware and load user program
- 4: reserved
- 8: as 0.. 4 + save debug data in case of possible failures
- 16: reserved

- 0: no update
- 1: update
- 2: update with specific version
- 3: update with specific version if the SD file is newer

- 0: no update
- 1: update

# SD Memory Card

## Store / Load a Project to the SD Card



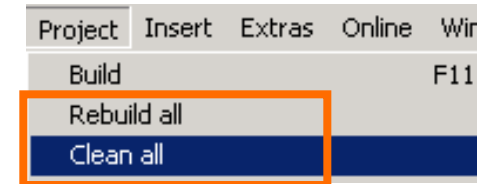
### Copy the project from PLC to PLC.


- Copy a project as a machine code not readable by PC and the configuration of communication modules to a SD card plugged in one PLC.
- Load this project into another PLC from the SD card plugged in there.

# SD Memory Card

## Store a Project to the SD Card (1)

- To store the user program to the SD card, proceed as follows:
  - Build the complete project using the menu items "Project" \ "Clean all" and "Project" \ "Rebuild all".
  - Download the project into the AC500.



 Caution: after download the program execution is stopped!

- Create the boot project on the controller using "Online" "Create boot project".
- The boot project files (DEFAULT.PRG and DEFAULT.CHK) are loaded into the AC500 and flashed. The "RUN" and "ERR" LEDs on the AC500 flash alternately while data flashing is in progress.

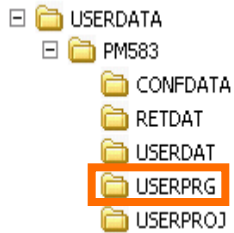
# SD Memory Card

## Store a Project to the SD Card (2)



- Open “Resources” / “PLC Browser”. Enter the command "**sdappl**"<ENTER>.
- The “RUN” LED on the AC500 flashes during writing to the SD card.

```
sdappl
sdappl
Writing user program and configuration data to SD card USERDATA\PM573ETH\USERPRG\
You might be logged off by CodeSys in between (e.g. logged in via CM577)
.
.
.
.
done.
```



- The files DEFAULT.PRG and DEFAULT.CHK and configuration files of the plugged communication modules are loaded from the flash memory and stored to the directory USERDATA\PM5xx\USERPROG on the SD card in the compressed file **BOOT.ZIP**.

[status]  
FunctionofCard

[FirmwareUpdat  
CPUPM5x1=0  
Display=0  
Coupler0=0  
Coupler1=0

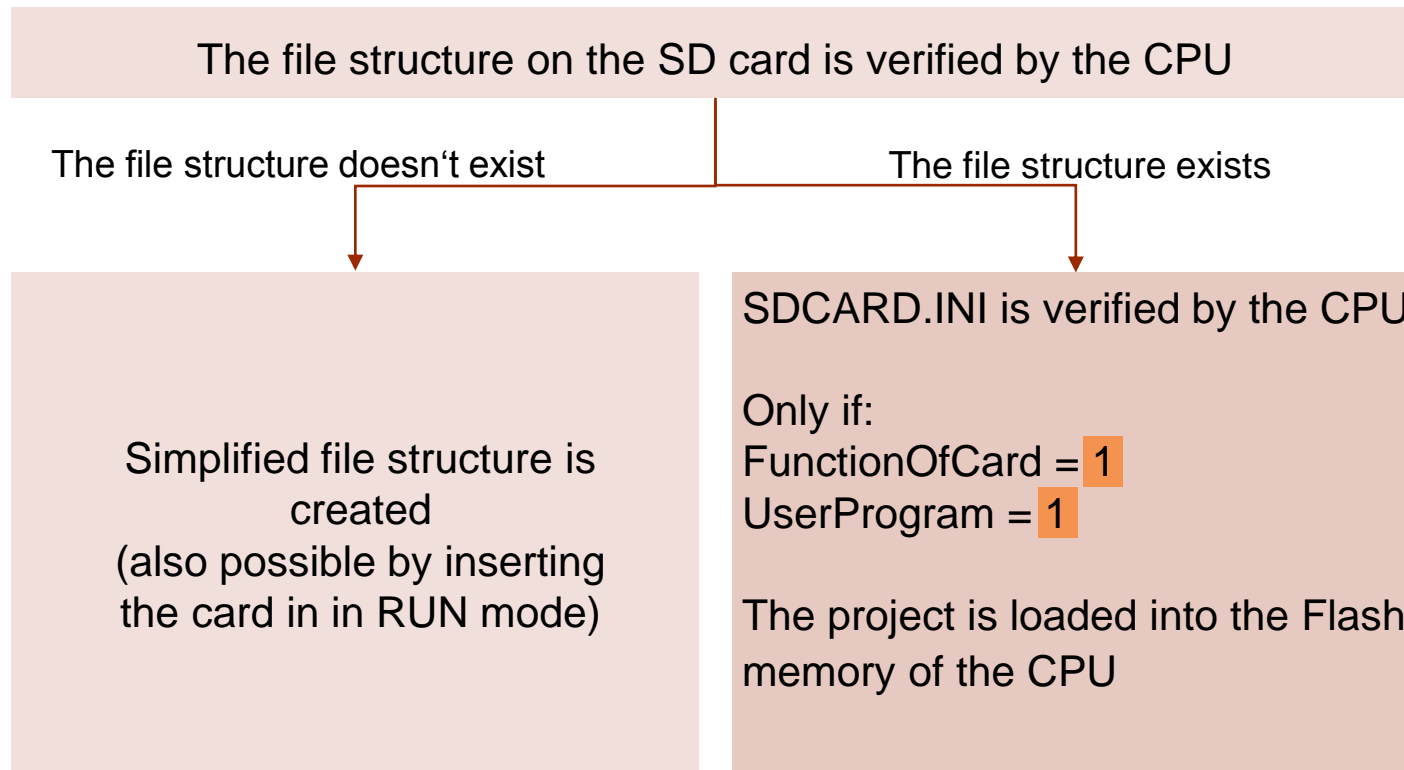
- In the file **SDCARD.INI** the parameters are set:
  - **FunctionOfCard = 1**
  - **UserProgram = 1**e.g., the function "Load the user program" is activated.

# SD Memory Card

## Load a Project from the SD Card

Due to duplicate a project, SD card has to be inserted into the AC500 when:

- The PLC is in STOP mode or
- Before switching the power ON



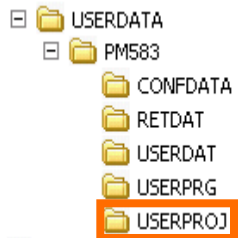
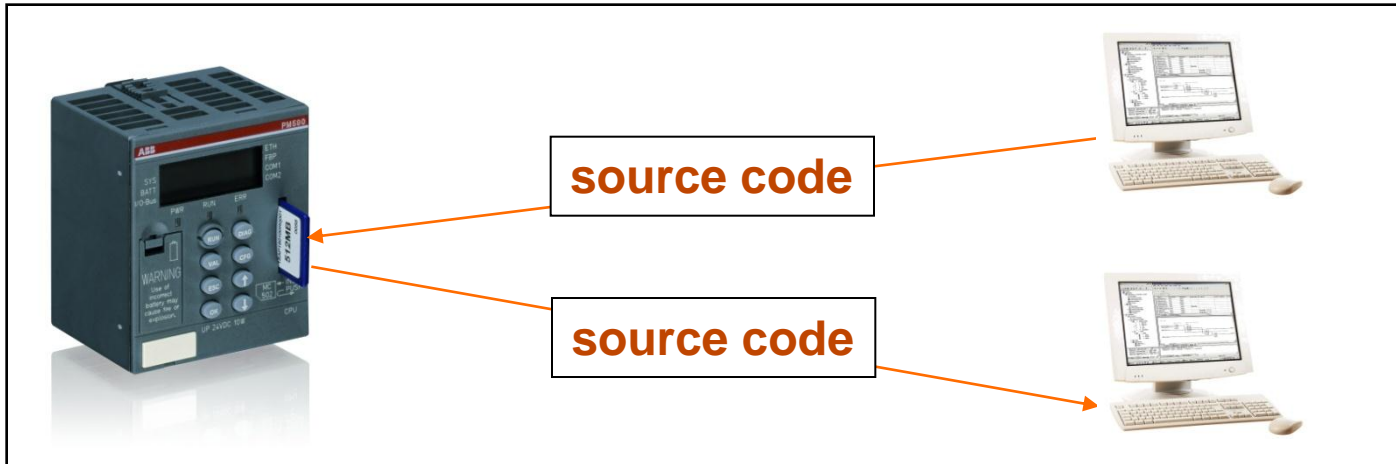
# SD Memory Card Coming up



- The file structure
- Store / load the project to / from the card
- **Source code download and upload**
- READ / WRITE data
- READ / WRITE retentive data
- Firmware update for PM5xx

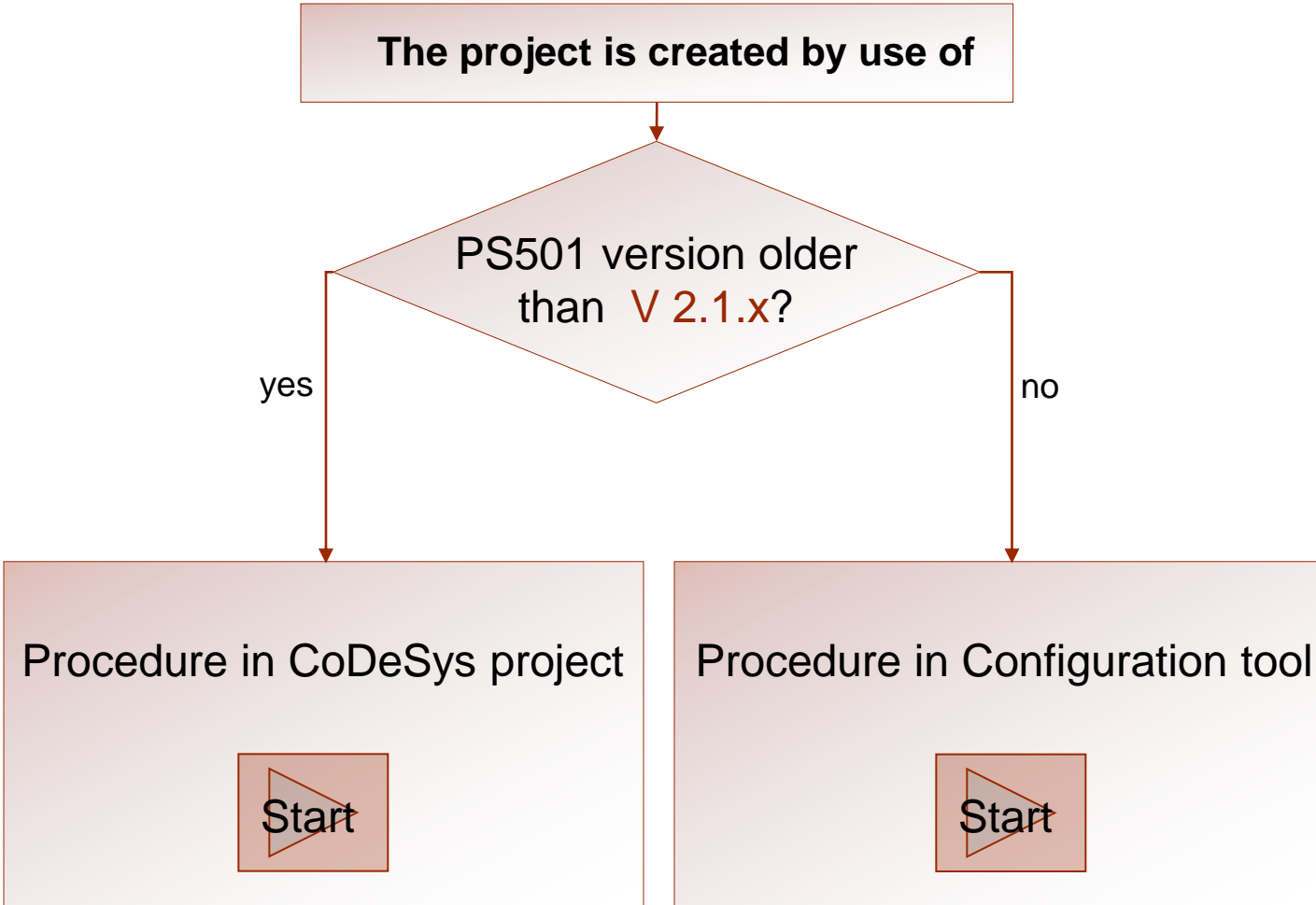
# SD Memory Card

## Source Code Download and Upload



- The whole PS501 project including comments, visualizations etc. stored in the PC can be stored as compressed file on the SD card plugged into the PLC.
- This backup can be uploaded from PLC or directly from SD card to any PC with installed PS501.

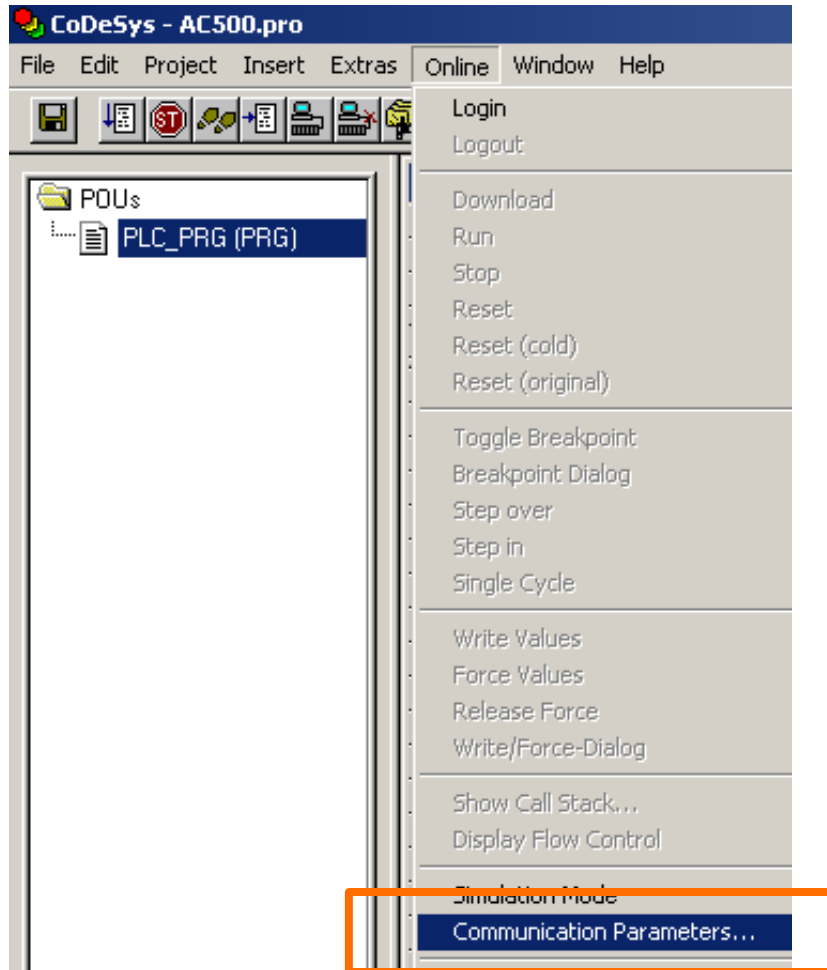
# SD Memory Card Source Code Download and Upload





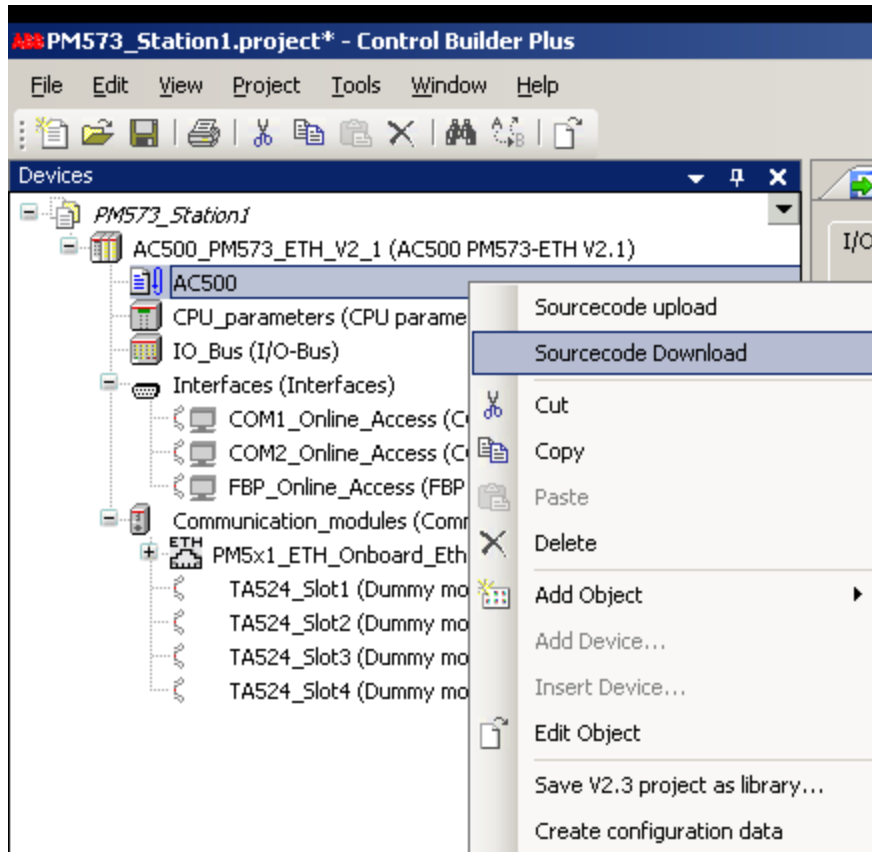
# SD Memory Card

## Source Code Download and Upload(1)

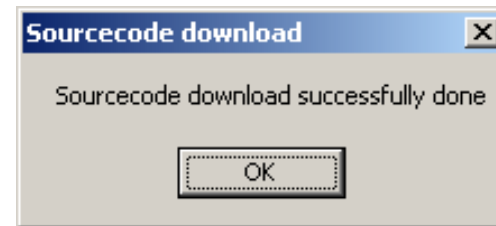


- Verify if the parameter “FunctionOfCard” on the SD card is set to “0” (no function when inserted).
- Verify if the SD card is plugged in the PLC.
- In **CoDeSys Project**: verify if the proper communication parameters are set. Save and exit the CoDeSys project.

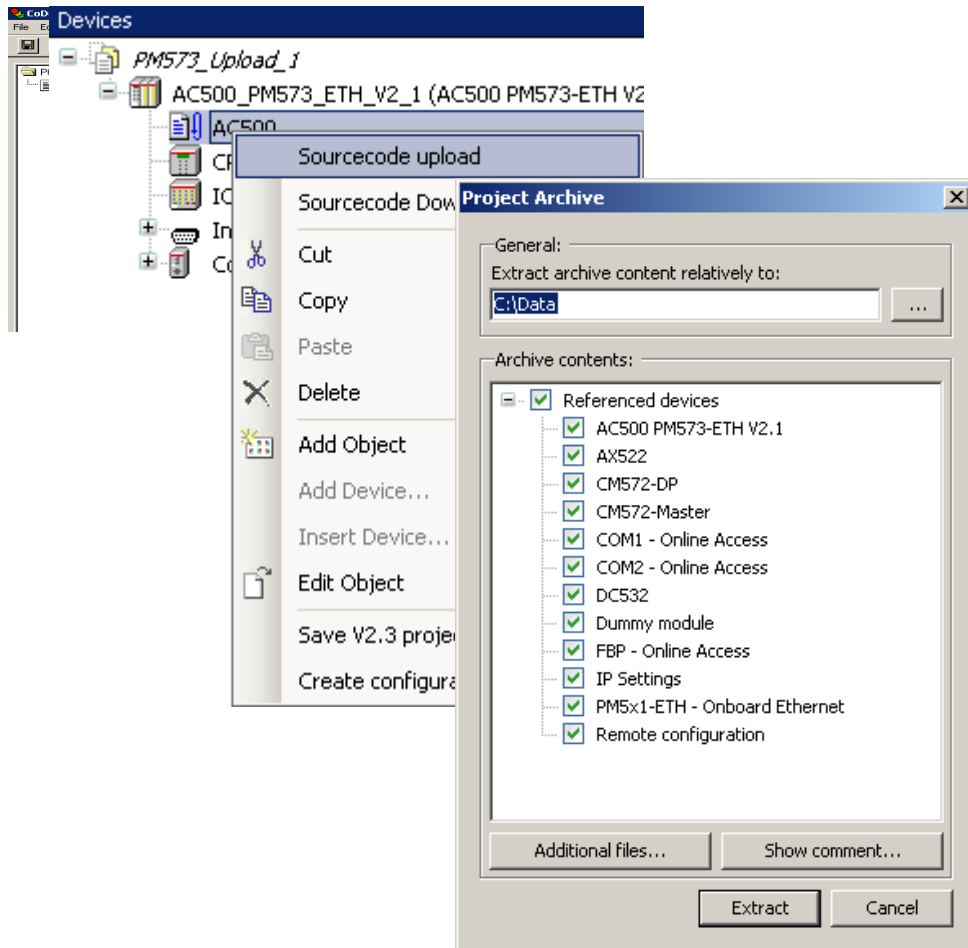
# SD Memory Card Source Code Download and Upload (2)



- In the **Configuration tool**:  
Start the procedure by right-click on the project branch and selecting the item “**Sourcecode Download**“. The CPU’s **RUN** led is blinking.
- The information about an successful download is given. On the SD-card the file SOURCE.ZIP has been created.



# SD Memory Card Source Code Download and Upload (3)



- Plug in the SD card into the slot on the CPU.
- Create a new project containing a CPU of the same type.
- In **CoDeSys Project**: verify if the right communication parameters are set or set them properly. Save and exit the CoDeSys project.
- In **Configuration tool**: right-click the project branch and select the item **Sourcecode upload**. The CPU's **RUN** led is flashing.
- Select the destination folder for the project and the files to be unzipped. It is recommended to unzip all listed files.

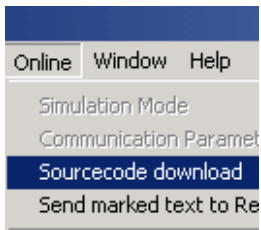
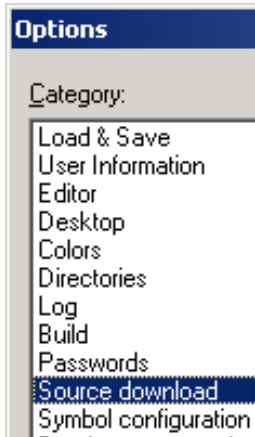
**Skip the action steps for version 1.x**



# SD Memory Card

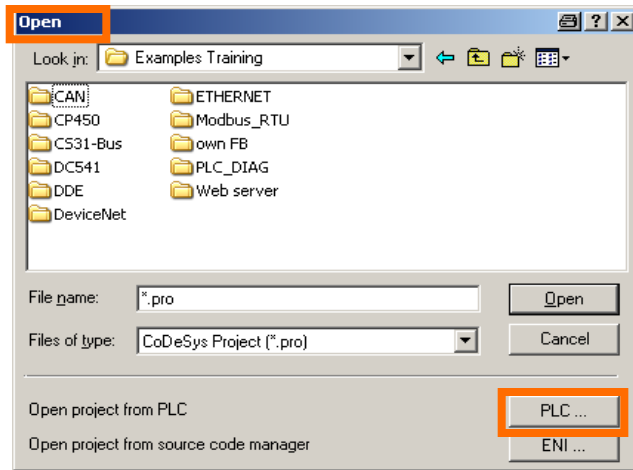
## Source Code Download to the SD Card

[Status]  
FunctionofCard

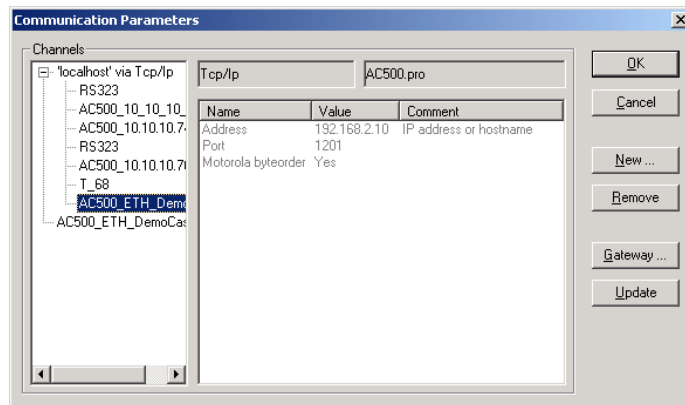


- Verify that the parameter “FunctionOfCard” on the SD card is set to “0” (no function when inserted).
- Verify that the SD card is plugged in the PLC.
- Define in PS501 when the download has to be done and what has to be downloaded:  
“Project” \ “Options” \ “Source download”.
- If “Only on demand” is selected, then you have to initiate the download by a click at “Online” \ “Source code download”.
- Download progress is indicated (it takes a longer time).

# SD Memory Card Source Code Upload from the PLC



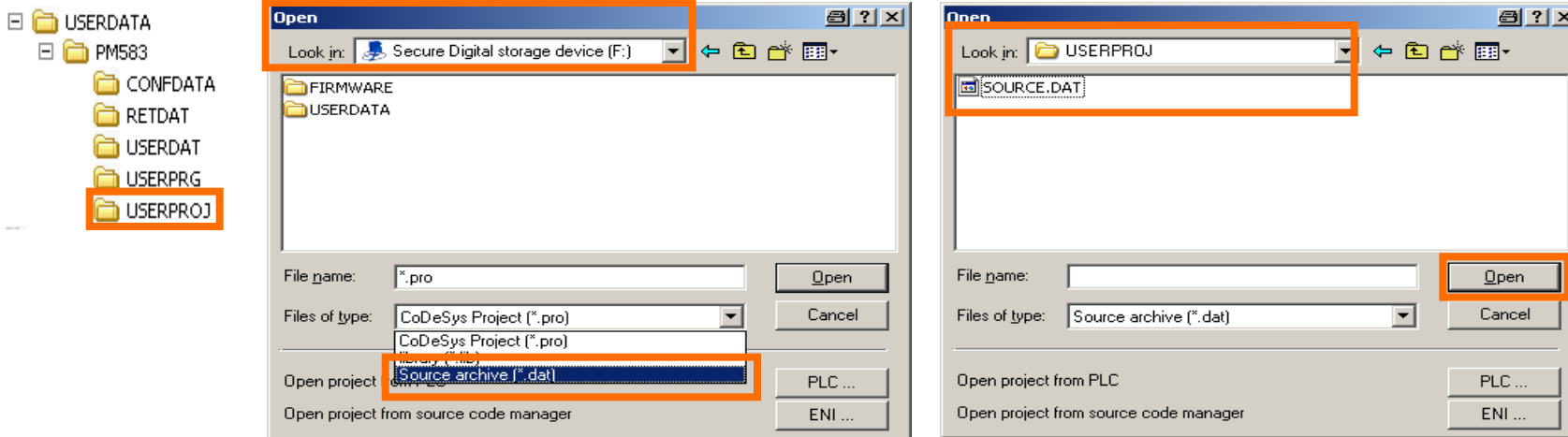
- Plug in the SD card into the slot on the CPU.
- Start PS501 and click at “File” \ ”Open“, then “PLC“.
- Select the CPU type from the list.



- Select the communication parameters for the data transmission.
- The upload progress is indicated (it takes a longer time).

# SD Memory Card

## Direct Source Code Upload from the SD card



- Plug in the SD card into the card reader on PC.
- Start PS501 and click at “File”\“open“.
- Select from the list the card reader directory.
- Select files of type “Source archive” (\*.dat).
- Open the file “Source.dat” in the appropriate path ...\USERDATA\PM5xx\USERPROJ.

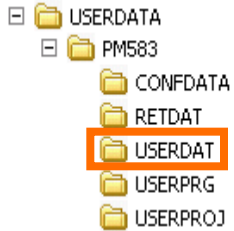
# SD Memory Card Coming up



- The file structure
- Store / load the project to / from the card
- Source code download and upload
- **READ / WRITE data**
- READ / WRITE retentive data
- Firmware update for PM581

# SD Memory Card

## Read / Write Data (1)



- Depending on the AC500 CPU type the data are stored in a file in following SD card directory path:  
**..\USERDATA\PM5xx\USERDAT\USRDATxx.DAT.**
- A maximum of 100 files (USRDAT00.DAT...USRDAT99.DAT) can be stored in one directory.
- Each data file USRDATxx.DAT can be divided into individual sectors, if necessary.
- The "sector label" enclosed in square brackets (such as [Sector\_01]<CR><LF>) indicates the start of the sector. Within a sector, data are saved as data sets in ASCII format.
- The individual values of a data set are separated by semicolon. Each data set is closed with <CR><LF> (0Dhex, 0Ahex).



# SD Memory Card Read / Write Data (2)

- This enables the direct import/export of the data from/to EXCEL.
- The data files can be viewed and edited by using a standard ASCII editor (e.g. Notepad).
- Example: File with 2 sectors, each with 3 data sets and 10 DINT values per data set:

```
[Sector_01]
610439;10408;10483;10446;10387;10442;10444;10452;10453;-1317225
610455;10423;10499;10462;10402;10458;10460;10467;10469;-1317209
610476;10445;10520;10483;10424;10479;10481;10489;10490;-1317188
[Sector_02]
610570;10539;10614;10577;10518;10573;10575;10583;10584;-1317094
610585;10554;10630;10592;10533;10589;10591;10598;10600;-1317078
610602;10571;10646;10609;10550;10605;10607;10615;10616;-1317062
```

# SD Memory Card Read / Write Data (3)

The screenshot shows a software help window with a table of contents on the left and a detailed view of the SD\_READ function block on the right. The table of contents includes:

- RETAIN\_RESTORE Write retain data from RAM-DISC to SRAM
- RETAIN\_SAVE Write retain data from SRAM to RAM-DISC
- RTS\_INFO Reading the version of the CPU runtime system
- Structure of the file USRDATXX.DAT on the SD Card
- SD\_READ Reading a data set from the SD Card**
- SD\_WRITE Writing a data set to the SD Card**
- SLOT\_INFO Reading slot information
- SYS\_TIME Reading the system time
- Glossary
- Index
- Modbus Library
- Series90 AC500 Library
- ASCII Communication Library
- CS31 Library
- Ethernet Library
- EtherCAT Library

The main window displays the title "SD\_READ Reading a data set from the SD Card" and a diagram of the function block. The diagram shows the following connections:

- SEG\_SD\_READ and ADRVAR\_SD\_READ connect to ADR inputs.
- EN\_SD\_READ, ATTRIB\_SD\_READ, FILENO\_SD\_READ, FORMAT\_SD\_READ, and NVAR\_SD\_READ connect to the SD\_READ block.
- SD\_READ block outputs: DONE, ERR, ERNO, and ADRVAR.
- Done\_SD\_Read, Err\_Sd\_Read, and Erno\_Sd\_Read are connected to the outputs.

- Library SysInt\_AC500\_V10.lib contains appropriate function blocks:
  - SD\_READ reads a data set from a file on the SD card and stores the read data set beginning at the start flag defined by ADRVAR.
  - SD\_WRITE writes a data set to a file USRDATxx.DAT on the SD card.

# SD Memory Card Coming up



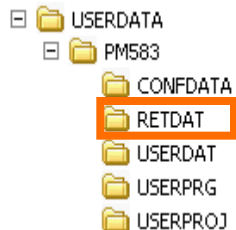
- The file structure
- Store / load the project to / from the card
- Source code download and upload
- READ / WRITE data
- **READ / WRITE retentive data**
- Firmware update for PM5xx

# SD Memory Card

## Read / Write RETAIN Data as of Version 1.2 (1)



```
PLC_PRG (PRG-FBD)
0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
0004 VAR RETAIN
0005   bVar1: BOOL;
0006 END_VAR
0007
0008
```



- Saving RETAIN data is done in two steps:
  - Copying the data from the RETAIN area and writing it to the CPU's RAM disk as file: **"save"**
  - Saving the file to the SD card: **"export"**
- Restore data from the SD card is done in two steps:
  - Loading the file from the SD card to the CPU's RAM disk: **"import"**
  - Copying the data from the RAM disk to the RETAIN area: **"restore"**
- The values of RETAIN data can be deleted by **"clear"**.
- Saving and restore the data can be done by means of:
  - The PLC Browseror
  - Function blocks contained in the library SysInt\_AC500\_V10.LIB.



# SD Memory Card

## Read / Write RETAIN Data as of Version 1.2 (2)

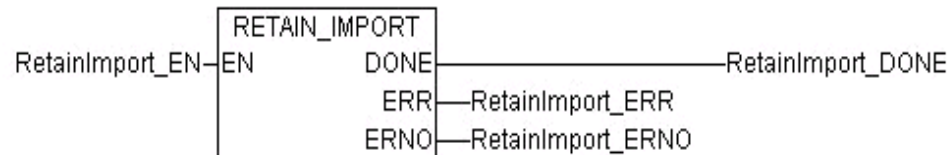
### PLC Browser:

retain	Saving and restoring the RETAIN variables: retain clear -> Clears all RETAIN variables retain save -> Saves the RETAIN variables to the RAM disk retain restore -> Restores the RETAIN variables from the RAM disk retain export -> Exports the RETAIN variables from the RAM disk to the SD card retain import -> Imports the RETAIN variables from the SD card to the RAM disk	Specific as of V1.2
--------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------

### Library SysInt\_AC500\_V10.LIB:

- ? PERSISTENT\_SAVE Write persistent data from SRAM to RAM
- ? PLC\_REBOOT Rebooting the PLC
- ? RETAIN\_CLEAR Delete retain data from SRAM
- ? RETAIN\_EXPORT Write retain data from RAM-DISC to SD Card
- ? **RETAIN\_IMPORT Write retain data from SD Card to RAM-DISC**
- ? RETAIN\_RESTORE Write retain data from RAM-DISC to SRAM
- ? RETAIN\_SAVE Write retain data from SRAM to RAM-DISC
- ? RTS\_INFO Reading the version of the CPU runtime system
- ? Structure of the file USRDATXX.DAT on the SD Card
- ? SD\_READ Reading a data set from the SD Card
- ? SD\_WRITE Writing a data set to the SD Card
- ? SLOT\_INFO Reading slot information

### RETAIN\_IMPORT Write retain data from SD Card to RAM-DISC



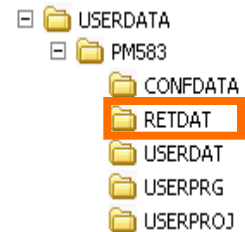
With the function block RETAIN\_IMPORT, all retain data can be written from the SD Card to the RAM-DISC.

# SD Memory Card

## Read / Write PERSISTENT Data up to Version 1.2 (1)

```
PLC_PRG (PRG-FBD)
0001 PROGRAM PLC_PRG
0002 VAR
0003   bVar1 AT %RX0.0:BOOL;
0004 END_VAR
nnnn
```

Auto run	On
Error LED	On
Check Battery	On
Behavior of outputs in st...	Off ir
Stop on error class	E2
Warmstart	Off
Reaction on floatingpoin...	E2 fe
Start PERSISTENT %R...	0
End PERSISTENT %RB...	100
Start PERSISTENT %R...	0
End PERSISTENT %RB...	0



- Saving buffered %R PERSISTENT data is done in two steps:

- Copying the data from the %R area and writing it to the CPU's RAM disk as file: "save"
- Saving the file to the SD card: "export"



- Restoring data from the SD card is done in two steps:

- Loading the file from the SD card to the CPU's RAM disk: "import"
- Copying the data from the RAM disk to the %R area: "restore"



- The buffered data of %R area can be deleted by "clear".

- Saving and restore the data can be done by using:

- The PLC Browser
- Function blocks contained in the library SysInt\_AC500\_V10.LIB.



# SD Memory Card

## Read / Write PERSISTENT Data up to Version 1.2 (2)

### PLC Browser:

Function	PLC Browser command	Function block
Copy from %R area to RAM disk	persistent save	PERSISTENT_SAVE
Save file to SD card	persistent export	PERSISTENT_EXPORT
Read file from SD card to RAM disk	persistent import	PERSISTENT_IMPORT
Copy data from RAM disk to %R area	persistent restore	PERSISTENT_RESTORE
Delete buffered data of the PERSISTENT area	persistent clear	PERSISTENT_CLEAR

### Library SysInt\_AC500\_V10.LIB:

**PERSISTENT\_EXPORT** Write persistent data from RAM-DISC to SD Card

```

    graph LR
      subgraph PERSISTENT_EXPORT [PERSISTENT_EXPORT]
        EN[EN]
        DONE[DONE]
        ERR[ERR]
        ERNO[ERNO]
        NUM[NUM]
      end
      EN --- DoneLine[ ]
      DoneLine --- PersistentExport_DONE[PersistentExport_DONE]
      ERR --- PersistentExport_ERR[PersistentExport_ERR]
      ERNO --- PersistentExport_ERNO[PersistentExport_ERNO]
      NUM --- PersistentExport_NUM[PersistentExport_NUM]
  
```

With the function block PERSISTENT\_EXPORT, all written data located in the PERSISTENT area or the %R area can be written from the RAM-DISC to the SD Card.

# SD Memory Card Coming up



- The file structure
- Store / load the project to / from the card
- Source code download and upload
- READ / WRITE data
- READ / WRITE retentive data
- **Firmware update for PM5xx**



# SD Speicherkarte

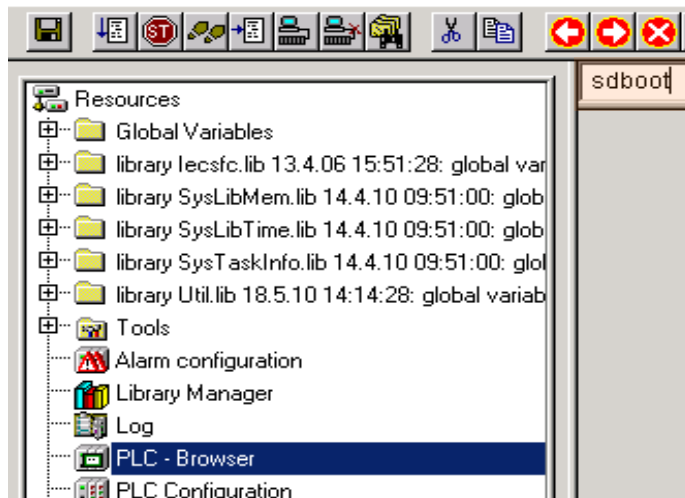
## Firmware Update for PM5xx (1)

Firmware update has to be performed in a well predefined order to avoid damage of the CPU:

1. step: Bootcode update
2. step: Firmware update

**Bootcode update** for all firmware versions

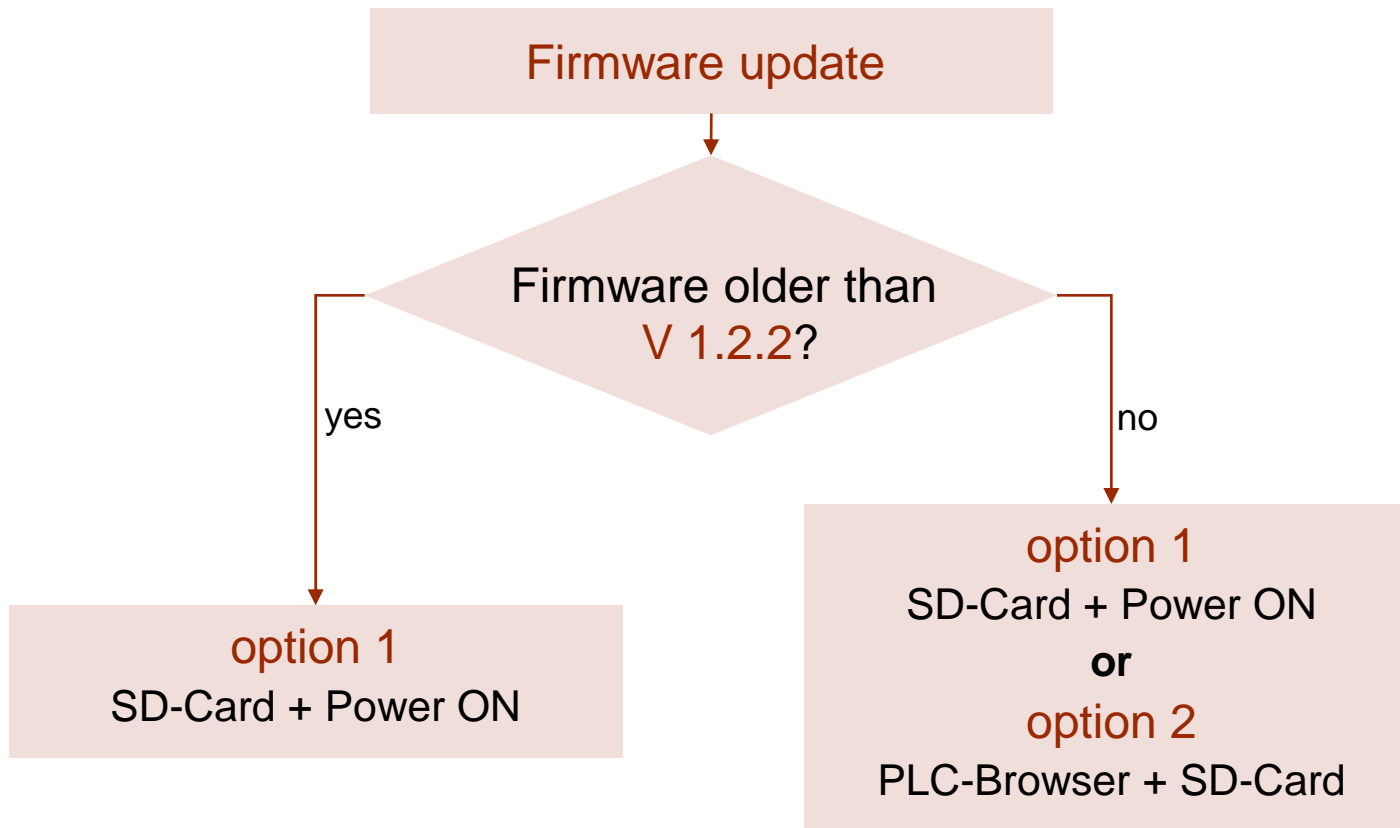
Tools: PLC-Browser + SD-card



- Insert the SD card containing the newest CPU firmware version into the CPU.
- Perform the boot code update using the PLC browser command **“sdboot”**.
- The boot code will be updated.

# SD Speicherkarte

## Firmware update for PM5xx (2)



# SD Memory Card Firmware Update for PM5xx

```
[status]
FunctionofCard=2

[Firmwareupdate]
CPUPM5x1=1
Display=0
Coupler0=0
Coupler1=0
Coupler2=0
Coupler3=0
coupler4=0
```

## Firmware update option 1:

- Set following parameters in the SDCARD.INI:

FunctionOfCard=**2**

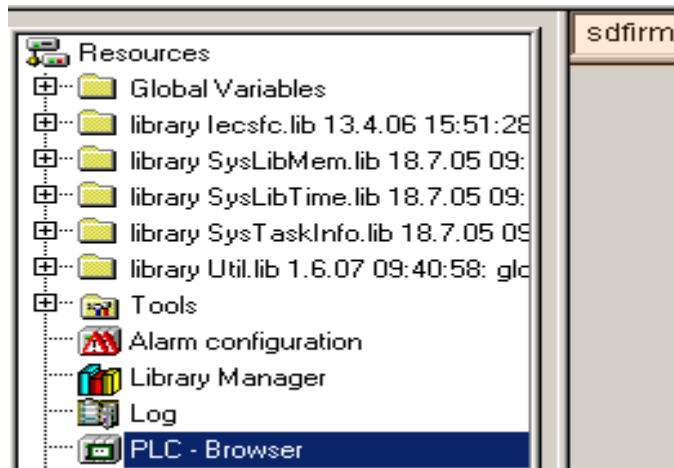
CPUPM5xx=**1**

- Plug the SD card into the CPU.

- Switch power OFF and again ON.
- The firmware update is done automatically and is indicated by “RUN” and ERR” LEDs.
- Do not switch power OFF while update is proceed!
- The update is done when the red “ERR” LED finished blinking and the green “RUN” LED is blinking slowly.

# SD Memory Card

## Firmware Update for PM5xx by use of PLC Browser



### Firmware update: option 2

- Plug the SD card into the CPU.
- Stop the program execution.
- Perform the firmware update using the PLC browser command “sdfirm”.
- The CPU firmware will be updated.

# SD Memory Card

## Cross-References to Documentation

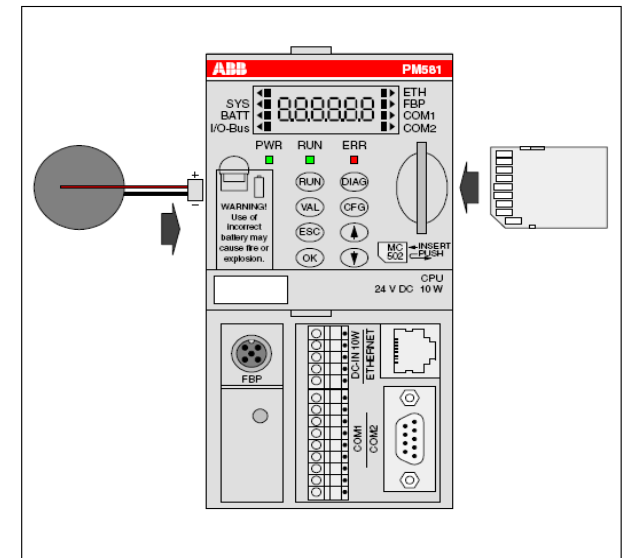
1. CoDeSys Help\Target System\AC500 / S500\System Technology\System Technology of the AC500 CPUs\The SD Memory Card in AC500
2. CoDeSys Help\Target System\AC500 / S500\Function Block Libraries AC500\Internal System Library\

System Description

**AC500**

Scalable PLC  
for Individual Automation

System Technology  
of the CPUs



Power and productivity  
for a better world™

