

PHYBOARD WEGA-AM335X

INTRODUCTION :

PHYBOARD-WEGA-AM335x

-->WegaBoard Supporting Processor

AM3352,AM3354,AM3356,AM3357,AM3358,AM3359 But This Board supports AM335X.

-->Wega-Board Architecture Based on "ARM Cortex-A8 ".

--> This Board Bitwidth from 32-Bit and frequency is up to 200 MHz to 1GHz.

--> Memory :- NAND Flash,eMMc,SPI NOR Flash.

--> Interfaces :- This Board supports Different Interfaces

Ethernet,USB,UART,Serial,CAN,I2C,SPI,MMC/SD,PWM,A/D,Display And Audio.

--> Physical Data :

Power supply	5v
Connectors Samtec	0.5 mm pitch (220 pins)
Dimensions	50 mm x 40 mm
Temperature Range	0 c to 70 c , -40 c to 85 c

---> Software :

Operating System	Linux (Yocto Based)
------------------	---------------------

Reference Using Below Link:

<https://www.phytec.de/produkte/system-on-modules/phycore-am335x/#technische-details/>

Am335x -Wega Board Supporting Images :

Barebox : This Board is Supporting to "Barebox.bin" Images is A Secondary Bootloader Instalazation of the HardWare and Kernel .

Barebox Configuration : Barebox.config

MLO Image :- This is Supporting for Peripheral Booting Devices and Adding Patches.

oftree Image :ubif:zImage :- This Images is Using Load the Kernal .

Image Generation for SD Card Flashing:-

The Below Steps follow How to Generate Images For To Flash the SD Card.

step 1 :- Click on the Phytex.de then Select the Products option

step 2 :- Click on Single Board Computer then choose PhyBoARD-Wega

step 3 :- Click on The Downloads then it Shows the Manuals & Documents ,Software And Design Files . Choose Software.

step 4 :- In the Software Choose Yocto -Linux Mainline-Kernel. Here The Developer

Choose the Version Based on the Wega -Board. for my Project I choose PD19.1.0

step 5 :- After Choosing the Version Click the Arrow Mark Which is on the Release Notes.

Then We Get link like Below

<https://download.phytec.de/Software/Linux/BSP-Yocto-AM335x/BSP-Yocto-AM335x-PD19.1.0/ReleaseNotes>

-->Remove ReleaseNotes from the above link

<https://download.phytec.de/Software/Linux/BSP-Yocto-AM335x/BSP-Yocto-AM335x-PD19.1.0/>

step 6 :- In the list Select Images After choosing Based on the Requirement of the Board.Select the Required Images.

step 7 :- hear I'm Considering The Below Image.

phytec-qt5demo-image-phyboard-wega-am335x-1.sdcard

step 8 :- Download the BOOT And Rootfs Images.

Partition:

In SD Card Make Two Partitions :1. Boot

2.Rootfs

--> Unmount all partitions with: host\$ umount /dev/<your_device><number>

--> After having unmounted all devices with an appended number

(<your_device><number>), you can create your bootable SD card with: host\$ sudo dd if=<IMAGE NAME>-<MACHINE>.sdcard of=/dev/<your_device> bs=1MB

conv=fsync

SD Card Partition Completed.

Bootting Methods :-

1) SD Flashing.

2) NAND Flashing .

SD Card Flashing :

Step 1: Insert the sd card on board.

Step 2: Boot switch is ON

Step 3: Give the power supply to that board.

Step 4: Stop the barebox level and follow the below commands Taken Exit Command.

Step 5: Click on the Source Commands.

Step 6: In The List choose MMC.

Step 7: SD Flash is Process is Done

\$ mount / dev /disk0.0 /mnt

or

\$ mount /dev/mmcblk0p1 /mnt/

\$ cd /mnt

\$ sh flash.sh

\$ reset

NAND Flashing :

Step 1: Stop the barebox level

\$ boot NAND

Step 2 : Remove the SD Card

Step 3 : Boot switch OFF

Step 4 : stop the barebox level

\$ edit /env/bin/init

```
$ -z "${global.boot.default}" ] && global.boot.default= net change to nand
$ Ctrl D
$ Saveenv
$ Reset
Step 5: Booting from Nand
```

Getting kernel source file :

_search "git.phytec.de" in browser and click on "linux-mainline" the WegaBoard supports "v4.14.78-phy" and "v4.14.78-rt47-phy" versions (refer release notes to know versions) check for the above versions
we have chosen "v4.14.78-phy" version's commit message"
we have downloaded tar file and extracted it.
we got "linux-mainline-4.14.78-phy1" i.e source code for kernel.

Downloading toolchain :

open phytec.de

products -> single-board-computer -> wega-board -> Downloads -> software -> Yocto-Linux Mainline-Kernel -> in PD19.1.1 Select "Release notes" -> in url remove Release-notes and search -> sdk -> copy .sh file url link

open the terminal at the source code path

```
$ wget https://download.phytec.de/Software/Linux/BSP-Yocto-AM335x/BSP-Yocto-AM335x-PD19.1.1/sdk/phytec-yogurt-glibc-x86\_64-phytec-qt5demo-image-cortexa8hf-neon-toolchain-BSP-Yocto-AM335x-PD19.1.1.sh
```

```
$ chmod +x phytec-yogurt-glibc-x86_64-phytec-qt5demo-image-cortexa8hf-neon-toolchain-BSP-Yocto-AM335x-PD19.1.1.sh
```

```
$ ./phytec-yogurt-glibc-x86_64-phytec-qt5demo-image-cortexa8hf-neon-toolchain-BSP-Yocto-AM335x-PD19.1.1.sh
```

after extracting click "enter"
then click "y"

then toolchain will be installed

to enable toolchain we need to give below command

```
. /opt/phytec-yogurt/BSP-Yocto-AM335x-PD19.1.1/environment-setup-cortexa8hf-neon-phytec-linux-gnueabi
```

Generating zImage by using kernel source file :

open source code path in terminal
enable toolchain
\$ am335x_phytec_defconfig
\$ make -j4
zImage will generated at "linux-mainline-4.14.78-phy1/arch/arm/boot " path copy zimage to sdcard and do flash

UART Loop back Test :

check the hardware manual regarding the uart pins,there are 4 default Uart pins

Expansion connector (x69) contains

uart0 - Tx- pin_12, Rx- pin_10

uart2 - Tx- pin_33, Rx- pin_31

uart3 - Tx- pin_36, Rx- pin_35

RS232 interface connector (x66) contains

uart1 - Tx- pin_5, Rx- pin_3, RTS- pin_4, CTS- pin_6

There are only two default uart ports are available

uart0 - ttyO0

uart1 - ttyO1

When booting boot it will take uart0 as default, when we short uart0 pins the board will not boot.

short uart1 pins and boot board

check uart1 port is available i.e ttyO1

then check loopback by using below command

```
$microcom -s 115200 /dev/ttyO1
```

UART Pinmuxing :

previous we had testeed for default uarts 0 and 1

Now we do pin mux for uart 2,3. By taking reference as uart1 add uart2 in "am335x-phycore-pcm-953.dtsi"

get pad names of uart 2 by using this link

<https://wiki.phytec.com/pages/viewpage.action?pageId=170527630>

with pad name get address by using this link

https://www.ti.com/lit/ug/spruh73q/spruh73q.pdf?ts=1709532507059&ref_url=https%253A%252F%252Fwww.google.com%252F

after adding uart2 pullup and pull down addresses and add uart2 status as "okay" then check the given addresses are already there in your file by using below command

```
$grep -nir "address" ./am335x*
```

If addresses are already used comment them

#save file

#enable tool chain

```
$make am335x_phytec_defconfig
$make -j4
#copy zimage and am335x-phycore-nand-eprom-rtc-spi-tmp.dtb file to sdcard
#remove offtree and rename am335x-phycore-nand-eprom-rtc-spi-tmp.dtb file to offtree
boot board
```

you will get ttyO2 port
follow same steps for uart3 you will get ttySO3 port

I2C pin muxing :

wega board supports three I2C - i2c0, i2c1, i2c2
I2C0 is default
pins in x71
pin15 - x_i2c0_scl
pin 16 - x_i2c0_sda

By taking reference as i2c-0 add i2c-1 in "am335x-phycore-som.dtsi"

get pad names of i2c-1 by using this link
<https://wiki.phytec.com/pages/viewpage.action?pageId=170527630>

with pad name get address by using this link
https://www.ti.com/lit/ug/spruh73q/spruh73q.pdf?ts=1709532507059&ref_url=https%253A%252F%252Fwww.google.com%252F

after adding I2C2 sda and scl addresses
Add I2C2 status as "okay"
Then check whether the given addresses are already there in your file by using below command

```
$grep -nir "address" ./am335x*
```

If addresses are already used comment them

```
#save file
#enable tool chain
$make am335x_phytec_defconfig
$make -j4
#copy zimage and am335x-phycore-nand-eprom-rtc-spi-tmp.dtb file to sdcard
#remove offtree and rename am335x-phycore-nand-eprom-rtc-spi-tmp.dtb file to offtree
#boot board
```

you will get i2c-1 port

follow same steps for i2c-2 you will get i2c-2 port

SPI pin muxing :

wega board supports two SPI - spi0, spi1
SPI0 is default

pins in x69

- pin 4 - cso (chip select)
- pin 5 - d1 MOSI
- pin 6 - do MISO
- pin 7 - clk (clock select)

By taking reference as spi0 add spi1 in "am335x-phycore-som.dtsi"

get pad names of spi-1 by using this link

<https://wiki.phytec.com/pages/viewpage.action?pageId=170527630>

with pad name get address by using this link

https://www.ti.com/lit/ug/spruh73q/spruh73q.pdf?ts=1709532507059&ref_url=https%253A%252F%252Fwww.google.com%252F

after adding spi1 cso, mosi, miso and clk

and add spi1 status as "okay"

then check whether the given addresses are already there in your file by using below command

```
$grep -nir "address" ./am335x*
```

if addresses are already used comment them

```
#save file
```

```
#enable tool chain
```

```
$make am335x_phytec_defconfig
```

```
$make -j4
```

```
#copy zimage and am335x-phycore-nand-eeeprom-rtc-spi-tmp.dtb file to sdcard
```

```
#remove ofttree and rename am335x-phycore-nand-eeeprom-rtc-spi-tmp.dtb file to ofttree
```

```
#boot board
```

you will get spi1 port in cd /sys/class/spi_master

wega images using yocto :

```
$ mkdir yocto_wega
$ cd yocto_wega
$ wget https://download.phytec.de/Software/Linux/Yocto/Tools/phyLinux
$ chmod +x phyLinux
$ ./phyLinux init
$ 1
$ 33
$ 4
#enable toolchain
$cd conf
$vi local.conf
#change machine name to "phyboard-wega-am335x-1"
$ bitbake -c compile linux-mainline
#source code generated
```

path of source code :

```
#enable tool chain
$cd tmp/work/phyboard_wega_am335x_1-phytec-linux-gnueabi/linux-mainline/4.14.78-phy6-r0.0/
```

path to dtsi file :

```
$cd /linux-mainline/4.14.78-phy6-r0.0/git/arch/arm/boot/dts
```

path to images :

```
$cd /linux-mainline/4.14.78-phy6-r0.0/build/arch/arm/boot/dts
```

pin muxing using yocto source code :

UART pin muxing:

```
#uart 2,3,4,5 code added to am335x-pcm-953.dtsi
```

```
uart2_pins: pinmux_uart2 {
    pinctrl-single,pins = <
        AM33XX_IOPAD(0x92c, PIN_INPUT_PULLUP | MUX_MODE1)    /*
mii1_tx_clk,uart2_rxd */
```

```

        AM33XX_IOPAD(0x930, PIN_OUTPUT_PULLDOWN | MUX_MODE1)  /*
mii1_rx_clk.uart2_txd */
        >;
    };

    uart3_pins: pinmux_uart3 {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x934, PIN_INPUT_PULLUP | MUX_MODE1)  /*
mii1_rxd3.uart3_rxd */
            AM33XX_IOPAD(0x938, PIN_OUTPUT_PULLDOWN | MUX_MODE1)  /*
mii1_rxd2.uart3_txd */
        >;
    };

    uart4_pins: pinmux_uart4 {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x920, PIN_INPUT_PULLUP | MUX_MODE3)  /*
mii1_rxd3.uart4_rxd */
            AM33XX_IOPAD(0x91c, PIN_OUTPUT_PULLDOWN | MUX_MODE3)  /*
mii1_rxd2.uart4_txd */
        >;
    };

    uart5_pins: pinmux_uart5 {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x944, PIN_INPUT_PULLUP | MUX_MODE3)  /*
mii1_rxd3.uart5_rxd */
            AM33XX_IOPAD(0x908, PIN_OUTPUT_PULLDOWN | MUX_MODE3)  /*
mii1_rxd2.uart5_txd */
        >;
    };

    &uart2 {
        pinctrl-names = "default";
        pinctrl-0 = <&uart2_pins>;
        status = "okay";
    };

    &uart3 {
        pinctrl-names = "default";
        pinctrl-0 = <&uart3_pins>;
        status = "okay";
    };

    &uart4 {
        pinctrl-names = "default";
        pinctrl-0 = <&uart4_pins>;
        status = "okay";
    };

    &uart5 {

```



```

    pinctrl-names = "default";
    pinctrl-0 = <&uart5_pins>;
    status = "okay";
};

```

I2C pin muxing:

#i2c0 and i2c1 code is added to am335x-phycore-som.dtsi

```

&am33xx_pinmux {
    i2c1_pins: pinmux_i2c1 {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x958, PIN_INPUT | MUX_MODE2) /* i2c0_sda.i2c1_sda */
            AM33XX_IOPAD(0x95c, PIN_INPUT | MUX_MODE2) /* i2c0_scl.i2c1_scl */
        >;
    };
};

```

```

&i2c1 {
    pinctrl-names = "default";
    pinctrl-0 = <&i2c1_pins>;
    clock-frequency = <400000>;
    status = "okay";
};

```

```

&am33xx_pinmux {
    i2c2_pins: pinmux_i2c2 {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x950, PIN_INPUT | MUX_MODE2) /* i2c0_sda.i2c2_sda */
            AM33XX_IOPAD(0x954, PIN_INPUT | MUX_MODE2) /* i2c0_scl.i2c2_scl */
        >;
    };
};

```

```

&i2c2 {
    pinctrl-names = "default";
    pinctrl-0 = <&i2c2_pins>;
    clock-frequency = <400000>;
    status = "okay";
};

```

SPI pin muxing :

#spi1 code is added to am335x-phycore-som.dtsi

```
&am33xx_pinmux {
    spi1_pins: pinmux_spi1 {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x990, PIN_INPUT_PULLDOWN | MUX_MODE3) /*
spi1_clk.spi0_clk */
            AM33XX_IOPAD(0x994, PIN_INPUT_PULLDOWN | MUX_MODE3) /*
spi1_d0.spi0_d0 */
            AM33XX_IOPAD(0x998, PIN_INPUT_PULLUP | MUX_MODE3) /*
spi1_d1.spi0_d1 */
            AM33XX_IOPAD(0x99c, PIN_INPUT_PULLUP | MUX_MODE3) /*
spi1_cs0.spi0_cs0 */
        >;
    };
};

&spi1 {
    pinctrl-names = "default";
    pinctrl-0 = <&spi1_pins>;
    status = "okay";
};
```