KHEPERA IV: Programming Requisites

Compiler:-

Khepera IV uses a cross complier instead of a native compiler for execution. A cross compiler is a compiler capable of creating executable code for a platform other than the one on which the compiler is running. For example, a compiler that runs on a Windows 7 PC but generates code that runs on Android smartphone is a cross compiler. In case of Khepera robot, the cross compiler helps to build a code on the computer running with Linux operating system and the execution of the code is done within the robot itself.

The cross-compiler used for this robot is **GCC (GNU compiler collection).** The GNU Compiler Collection (GCC) is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU toolchain and the standard compiler for most Unix-like operating systems.

Libraries and Preprocessors

Khepera IV API include all the libraries after installing the cross-compiler in /opt/poky folder. The declaration of all in build khepera functions are inside **"khepera/khepera.h".**

Basic Functions

An important variable must be discussed which is used in almost all programs is **dsPic**. It is a pointer variable utilized for microcontroller access. It is of a data type named **knet_dev_t**. Moreover, it is a static variable. All other important functions are described in the table below:-

Function name	Input type	Output type	Remarks	Return
kb_clrscr	Void	void	Clear the console screen	none
kb_change_term_mode	int *dir*	void	Change terminal mode for getchar to return immediately dir 1= mode changed to non-blocking, 0 mode reverted to previous	Returns none
ctrlc_handler	int sig	void	static	Returns none
kb_kbhit	void	integer	Test if anykey was pushed	Returns -1 if error occured >=0 number of characters to read
kh4_get_speed	int * *left*, int * *right*, knet_dev_t * *hDev*	integer	kh4_get_speed get motors speed left motor speed (units: encoder) right motor speed (units: encoder) It is a handle to an opened knet socket (Khepera4:dsPic).	Returns A value: <0 on error >=0 on success
kb_change_term_mode	int *dir*	void	Change terminal mode for get char to return immediately Dir 1= mode changed to non-blocking, 0 mode reverted to previous	Returns none
main	int argc, char *argv[]	*integer*	Provide a basic test program to control a K-Team pan tilt camera with a KoreMotor using the KoreBot library.	Returns none
kh4_init	int *argc*, char * *argv*[]	integer	kh4_init initializes some things like the GPIO40 pin. This function needs to be called BEFORE any other functions.	Returns A value: <0 on error =0 on success

Function name	Input type		Output type	Remarks	Return
knet_open	const char *	*device*,	void knet_dev_t*	This function opens a K-Net device given its name K-Net Device name to open	A pointer to a K-Net device descriptor or NULL if no device is found Remarks This function is thread safe.
	int	*preferred_bus*,		Preferred Bus.	
	int	*argc*,		argument count	
	char *	*argv*[]		argument vector	
kh4_SetPositionMargin	int	*margin*,	integer	kh4_SetPositionMargin set position margin for control	Returns A value: <0 on error >=0 on success
	knet_dev_t *	*hDev*		hDev is a handle to an opened knet socket (Khepera4:dsPic).	
kh4_SetSpeedProfile	char	*accinc*,	integer	kh4_SetSpeedProfile configure motor control speed profile Acceleration increment	Returns A value: <0 on error >=0 on success
	char	*accdiv*,		Acceleration divider	
	char	*mindacc*,		Minimum speed acc	
	char	*mindec*,		Minimum speed dec	
	int	*max_speed*,		maximum speed	
	knet_dev_t *	*hDev*		It is a handle to an opened knet socket (Khepera4:dsPic).	
kh4_revision	char *	*outbuf*,	integer	It is a buffer where the data will be stored on.	Returns A value: <0 on error >=0 on success
	knet_dev_t *	*hDev*		It is a handle to an opened knet socket (Khepera4:dsPic).	
Take_one_image	Unsigned char*	*buffer*	integer	Taking an image in rgb format and storing in buffer	Returns A value: >=0 on success <0 on error
Save_buffer_to_jpg	Const char*	filename	integer	Filename of the image taken	Returns A value: >=0 on success <0 on error
	int	quality		Jpeg quality in range [0-100]	
	Unsigned char*	buffer		Image buffer	

Function name	Input type		Output type	Remarks	Return
kb_camera_release	void	none	void	Release the camera and the device	Returns A value: >=0 on success <0 on error
kb_camera_init	unsigned int*	width	integer	This function needs to be called in the beginning of image processing mechanism. The width and height denotes the image's dimensions.	Returns A value: >=0 on success <0 on error
	Unsigned int *	height			
into_grayscale	unsigned char *	src		Convert image to grayscale	Returns A value: >=0 on success <0 on error
apply_filter	unsigned char*	src	integer	Source of image	Returns A value: >=0 on success <0 on error
	unsigned char*	dst		Destination of processed image	
	int *	filter_x_array		Filter square array for column dimension	
	int *	filter_y_array		Filter square array for line dimension	
	int	filter_dim_x		Size of filter for column dimension (must be odd)	
	int	filter_dim_y		Size of filter for line dimension (must be odd)	