USBTiny-MKII programozó

Software telepítés

Miután elkészítettük a programozónkat és az hibátlanra sikerült (forrasztások, összes átkötés – via – megléte, szemrevételezés legalább 12x-s nagyítóval, különös tekeintettel a GTL szintillesztő lábai közötti terület ellenőrzése, stb ...) elméletileg azonnal működőképes állapotban kell lennie. Ezért egy USB kábel segítségével azonnal csatlakoztatjuk a számítógéphez. Mivel még egy szűz processzorról van szó, ezért maga az AT90USBxxx jelentkezik. Mivel a gépnek fogalma sincs, hogy mivel van dolga, ezért ezt látjuk az eszközkezelőben: ismeretlen eszköz. Viszont ez egy nagyon jó és megnyugtató hír. Mivel így már biztosak lehetünk benne, hogy a processzor működik és az USB rész szintén, mivel létre

Billentyűzet
 Bluetooth-rádiók
 DVD/CD-ROM-meghajtók
 Egér vagy egyéb mutatóeszköz
 Egyéb eszközök
 Egyéb eszközök
 AT90USB162 DFU
 Hálózati kártyák
 Hang-, video- és játékvezérlők
 IDE ATA/ATAPI vezérlők

jött a kapcsolat a programozó és a PC között. Ez már fél siker, a fele már működik. Ilyenkor az egyik LED-nek sem kell világítania. Ha netán a Win szeretné a meghajtó software-t telepíteni, ne hagyjuk neki, mert teljesen felesleges. Feltéve, ha ez az első telepítésünk, mert különben úgyis felismerte volna. A következő lépés, hogy telepítjük a FLIP-t, amivel életet lehelünk a programozónkba. Ez egy JAVA alapú programozó

software, amivel az összes nativ USB-s Atmel processzort tudjuk programozni, ugyanis ezekbe már gyárilag található egy bootloader. Itt már egyébként láthatjuk a bootloader-t működés közben, mivel már felépített egy kommunikációt a PC-vel. Ezért tudja a processzor típusát. Most letöltjük a programot innen: <u>http://www.atmel.com/tools/FLIP.aspx</u>. Kiválasztjuk a nekünk megfelelőt. Ha nem vagyunk biztosak benne, hogy a JRE már telepítve van a gépünkön, akkor ezt **"FLIP 3.4.7 for Windows (Java Runtime Environement included)"** töltjük le. Kezdjük el a telepítést. Ha a JRE nincs telepítve, ezt látjuk:

G Application Setup	
Java Runtime Environment Installation	
No Java Runtime Environment could be found on your computer. Click Install to start the installation of JRE 1.5.0.	
Nullsoft Install System v2.26	Cancel

Végig megyünk a telepítésen. Viszont, ha a teljes csomagot telepítjük, vagyis JRE-vel együtt, akkor mindenképpen frissítsük a JRE-t, mert ez már nem a legfrissebb. És érdekes módon, ha a JAVA a háttérben frissít, a Flip hajlamos meglepetésszerűen kilépni. Frissítés után stabil. A telepítés után elballagunk ide:



Itt találjuk ugyanis az USB meghajtót, hogy végre működjön az ügy. Ide csak azért jöttünk, hogy lássuk, honnan kell telepíteni a meghajtót. Az eszközkezelőben jobb klikk az ismeretlen eszközre AT90USBxxx, majd illesztő program frissítése. Itt kiválasztjuk, hogy mi keressük meg az illesztő programot. Tehát kijelöljük a C:\Program Files\Atmel\Flip 3.4.7\usb könyvtárat és hagyjuk, hogy telepítse az illesztőt. Ha ez megtörtént, ezt kell látnunk az eszközkezelőben.

Atmel USB Devices
 AT90USB162
 Billentyűzet
 Bluetooth-rádiók
 DVD/CD-ROM-meghajtók
 Bgér vagy egyéb mutatóeszköz

Innentől kezdve, már elérhető lesz a Flip számára is. Most már fel tudjuk tölteni az általunk kiválasztott firmware-t a programozónkra. Itt már látható, hogy talán egy kicsit bonyolultnak tűnő előkészítés után már nagyon egyszerű dolgunk lesz. És valóban a firmware csere nagyon gyors. Nem beszélve arról, hogy már látható, hogy ha valaki egy

ilyen nativ USB processzorral szerelt áramkört tervez, egyáltalán nincs szükség programozóra. De most hívjuk fel a Flip-t. Persze az áramkörünk továbbra is ott figyel az USB porton. Az első dolgunk, hogy rákattintunk a Chip ikonra, a bal felső sarokban "Select Target Device" és a megjelenő ablakban kiválasztjuk a programozandó chip típusát. Hogy miért nem ismeri fel magától, az számomra egy rejtély. Nem hiszem, hogy rettenetesen nagy problémát jelentett volna, de hát a slamposságnak nincs határa.



Ezután a kommunikációs vonalat kell kiválasztanunk. Ez az USB kábeles ikon. Itt értelemszerűen az USB-t választjuk. Ezek után megnyitjuk a csatornát és ezt látjuk:

🚮 Atmel Flip		
File Buffer Device Sett	ings Help	
🤝 😴 🧔	ilia ilia ilia ilia ilia ilia ilia ilia	🐚 🏄 🛃
Operations Flow	FLASH Buffer Information	AT90USB162
🚳 🔽 Erase	Size 12 KB	Signature Bytes 58 1E 94 82
	Range 0x0 - 0x0	Device Boot Ids 00 00
	Checksum 0xFF	, ,
🕥 📄 Blank Check	Reset Before Loading	Bootloader Ver. 1.0.5
🕥 📝 Program	HEX File:	
🕥 🕡 Verify		
Run	Select EEPROM	Start Application 🔽 Reset
		USB ON

Láthatólag mindent felismert, amit csak kell. A következő lépés az általunk választott hex file, vagyis a firmware betöltése. File -> Load HEX file ... vagy a megfelelő ikonra klikkelve betöltjük a HEX-t.

Ezt kell látnunk.

📶oads\USBTiny MKII\co	mpiled-130303\AVRISP-MKII_130303	3_toggle_driver.hex
File Buffer Device Setting	gs Help	
🤝 😴 🧔	il 👫 畅 🕹	🔄 🏄 🏄
Operations Flow	FLASH Buffer Information	AT90USB 162
🔘 🔽 Erase	Size 12 KB	Signature Bytes 58 1E 94 82
	Range 0x0 - 0x29B5	Device Boot Ids 00 00
	Checksum 0x148E03	
🚳 🔲 Blank Check	Reset Before Loading	Bootloader Ver. 1.0.5
🕥 📝 Program	HEX File: AVRISP-MKII_130303_toggle_driver.l 10678 util bytes	
Verify		
Run	Select EEPROM	Start Application 📝 Reset
HEX file parsed.		USB ON

Aztán klikk a Run-ra. A programozás után minden zöld.



Ezután klikk Start Application ikon. Ezzel indítja a programot a processzorban. Ha az AVR Studio még nem lett telepítve, azt tegyük most meg. A programozót a telepítés alatt válasszuk le a PC-ről. A telepítés után csatlakoztassuk a programozót. A Reset gombját ne nyomkodjuk. Később bővebben, hogy miért. Azonnal felismeri, hogy új hardware és elkezdi telepíteni az illesztő programot. Ha kész van, így néz ki az eszközkezelő:



Már fel is ismerte. Ez már igazi sikerélmény. És már látjuk, hogy ég az egyik LED. Ez a LED mindig ég, ha fut a programozó software. Indítsuk el az Atmel Studiot. A menüpont View -> Available Atmel Tools alatt láthatjuk fáradozásaink eredményét.



Ez azért már valami. És most egy gyors teszt. Viszont még játszunk egy kicsit, hogy megértsük mit is csinálunk és mink van. Nyomjuk meg a programozón a RESET gombot és nézzük mereven a képernyőt.

Available Tools	→ 🗆 ×
Tools and Simulators	Status
AVRISP mkII (000200212345) Program only	Disconnected
AVRISP mkII (000200312345) Program only	Connected
Simulator	Connected
F	
Ĵ	

A 212345-s eszközt leválasztotta és csatlakoztatta a 312345-s eszközt. Vagyis mostantól a másik firmware él. Mégpedig a LibUSB-s firmware. Tehát összefoglalva:

AVRStudio Jungo Driver	-> 212345	startnál 2x villog
AVRDude, BASCOM, LINUX LibUSB Driver	-> 312345	startnál 4x villog

Nyomjunk még egyszer egy RESTET, hogy vissza térjünk a Jungo Driver-hez. Ha netán a Studio alatt véletlenül a LibUSB-s firmware-el szeretnénk emberkedni, nem történik semmi, csak hibajelentést kapunk. Tehát ne a hardware-ünkben keressük a hibát. Most pedig vegyünk elő valamilyen áramkört, amin tesztelhetjük az áramkörünk másik oldalát. Az eddigi visszajelzések szerint itt szokott problémás lenni az ügy. Az egyik Forum társunk majd egy hétig kereste a hibát, ami végül is egy hidegforrasztás volt. Persze, hogy a GTL-nél. Ezzel csak arra szeretném felhívni mindenki figyelmét, ha netán nem működne, hideg fejjel és módszeresen zárjunk ki mindent ami jó és ami marad, ott van a kutya elásva. Én egy próbapanelen felépített tranzisztor tesztert kötöttem rá, ami egyébként is a kísérletező kedvemnek esett áldozatul.



Aztán elballagunk az AVRStudioba, és ami most jön, az verziófüggő. Menü pont Tools -> Device programming. A felugró ablakban beállítjuk a programozónk típusát, a céláramkörön található AVR chip típusát, majd Apply.

Device Programming	1-2-1	12.2.2.8	Install	8 23
Tool Devia AVRISP mkII 🔻 ATr	ce Interfac mega328P v ISP	ce De	evice signature	Target Voltage
	Firmware Upgrade	re must be update	d before continuing	×
	Firmware Version	On Tool 1.14	On Disk 1.16	
	Firmware Upgrade			
			Upgrade Clo	se
				Close

Na, ez konkrétan úgy hiányzott, mint hátunkra a púp. Frissíteni ugyanis innen nem tudunk. Ez az eredeti Atmel programozónak szól, ami teljesen más processzort használ. Persze ha vicces kedvünkben vagyunk, nyomhatunk egy Upgrade-t, amire elkezd méltatlankodni, amúgy nem történik semmi, de tovább sem enged. Ezt ezzel a verzióval csinálja:

About AtmelStudio	×
	STUDIO 6
Atmel Studio 6 (Version: 6.1.2562) http: © 2013 Atmel Corp. Support: http:// All rights reserved. Support: http://	//www.atmel.com/ /support.atmel.com
OS Version: Microsoft Windows NT 6.1.7601 Service Pack 1 (Platform: Win	32NT)
View Installed Components: Shell VSIX manifest	
Installed Packages: Shell VSIX manifest - 6.1 Shell VSIX manifest (6.1)	Copy to clipboard
Package details: Shell VSIX manifest Version: 6.1 Package GUID: 5aa6ea3e-da7b-48c1-9b2a-cab2329d32ac Company: Atmel Coproration	Atmel
	ОК

Ha régebbi verziótok van, azzal feltehetőleg megy. Ez valószínűleg azzal lehet összefüggésben, hogy a firmware 2013 márciusi verzió, az Atmel meg utána hozta ki ezt a software csomagot. Tehát le kell erről az upgrade mániáról szoktatnunk. Ezért először elballagunk Vezérlőpult -> Mappa beállításai, majd Nézet fül és kivesszük a pipát az "Ismert fájl típusok kiterjesztésének elrejtése" elöli négyzetből. Ezután ide megyünk:



Majd átnevezzük pl.: avrispmkii.zip_, hogy még véletlenül se találja meg. Ez ugyanis a gyári programozóba való firmware. Most vissza a Studio-ba és újbóli próbálkozás.

AVRISP mkII (0002002123	345) - Device Prog	gramming		. 1.00	a transfer to the	? ×
Tool Devic AVRISP mkII	e 1ega328P ▼	Interface ISP • Apply	Device signature 0x1E950F	Read	Target Voltage 3,3 V Read	۵
Interface settings Tool information Device information Memories Fuses Lock bits Production file	ISP Clock	frequency must be lowe	er than 1/4 of freque	ncy the de	vice is operating c	125 kHz on. Set
Getting VtargetOK	ОК					
						Close

Ha ezt látjuk, jó a programozónk. A Device signature -> Read és Target Voltage -> Read gombot azért nyomjuk meg, hogy lássunk is valamit. A gyári programozó méri a céláramkör feszültségét. A miénk nem. A 3,3V csak azért van hogy a Studio ne izguljon. Mindig ennyi, legyen az a valóságban bármi is. Most viszont csináljunk is valamit.

AVRISP mkII (0002002123	2345) - Device Programming) x
Tool Devic AVRISP mkII	ce Interface Device signature Target Voltage mega328P ISP Apply Ox1E950F Read 3,3 V Read	2
Interface settings Tool information	Device Erase Chip Erase now	
Device information	Flash (32KB)	
Memories	C:\Users\Pluto\Downloads\V107K_EN_M168_240413\TransistorTester.hex	•
Fuses Lock bits	Image: Second	ad
Production file	EEPROM (1KB) C:\Users\Pluto\Downloads\V107K_EN_M168_240413\TransistorTester.eep	·
	Verify EEPROM after programming Program Verify Res	ad
Getting VtargetOK		
💽 ОК		
		lose

Kiválasztjuk a HEX-t amit be szeretnénk égetni a céláramkörbe, ha van EEPROM fájlunk akkor azt is és kattintunk a Program gombra.

AVRISP mkII (000200212	345) - Device Programming	? ×
Tool Devi AVRISP mkII	ce Interface nega328P V ISP V Apply	Device signature Target Voltage 0x1E950F Read 3,3 V Read
Interface settings Tool information	Device Erase Chip ▼ Erase now	
Memories	C:\Users\Pluto\Downloads\V107K_F	N_M168_240413\TransistorTester.hex
Fuses Lock bits	 Erase device before programming Verify Flash after programming 	9 Program Verify Read
Production file	EEPROM (1KB) C:\Users\Pluto\Downloads\V107K_{	N_M168_240413\TransistorTester.eep
	Verify EEPROM after programmir	ng Program Verify Read
Erasing device OK Programming FlashOk Verifying FlashOK	ç	
Verifying Flash.	OK	
		Close

Ha ezt látjuk, hátradőlünk és megiszunk egy sört! Ugyan ez az EEPROM fájlal is, ha van. És ezzel készen is vagyunk a Jungo-Driver résszel. Itt végül is akár le is állhatnánk, mivel bármivel is készítünk HEX-t, ha a Studio nyitva van, egyszerűen be tudjuk égetni a segítségével. Mi viszont tovább megyünk. Mindenek előtt kell egy LibUSB driver. Innen tudjuk letölteni: <u>http://sourceforge.net/projects/libusb-win32/?source=directory</u> Ha megvan, kicsomagoljuk. A programozó mindenképpen legyen csatlakoztatva a PC-re és a RESET gomb megnyomásával váltsunk át a LibUSB-s firmware-re. Ellenőrizzük, hogy az átkapcsolás után valóban 4x villognak a LED-k. Ha ez megvan, felhívjuk ai inf-wizard.exe-t, és végigmegyünk a telepítésen.

Letöltések ► libusb-win32-bin-1.2.6.0 ► bin ►			
iyitás Megosztás ezzel 🔻 Írás Új mappa			
Név	Módosítás dátun		
🐌 amd64	2012.01.17.8:40		
🄑 ia64	2012.01.17.8:40		
🍑 x86	2012.01.17.8:40		
inf-wizard.exe	2012.01.17.8:40		
libusb-win32-bin-README.txt	2012.01.17.8:39		

Ha megvagyunk, az eszközkezelőben ezt látjuk:



A programozónk "kijött" a Jungó Driver alól és megjelent az "Atmel USB Devices" alatt. Így már tudjuk használni LINUX és AVR Dude alatt is. És most ugrik a majom a vízbe, mert idáig a BASCOM nem mindegyik firmware verzióval volt hajlandó együttműködni. Most ez egy demo verzió:

About	×
	BASCOM-AVR
Compiler version Compiler build IDE version Serial number	: 2.0.7.5 : 2.0.7.5.003 : 2.0.7.5.003 : Serial DEMO
Windows OS Windows SP Company Owner Windows dir App data dir XML data dir System dir	<pre>Windows 7 Ultimate : Service Pack 1 : : Pluto : C:\Windows : C:\Users\Pluto\AppData\Local\M : C:\Users\Pluto\AppData\Roaming : C:\Windows\system32</pre>
Support	: <u>support@mcselec.com</u>

Tehát minden erre vonatkozik. Régebbi verziókkal nem teszteltem. Legelőször megnézzük a programozónkat az Options -> Programmer menüpont alatt.

BASCOM-AVR Optior	15	
Compiler Communica	ation Environment Simulator	Programmer Monitor Printer
Programmer	USBprog Programmer / AVR IS	SP mkll 🔹
Play sound		
Erase warning	mpile	AutoVerify Upload Code and Data Set focus to terminal emulator after programming
Atmel		
COM-port (COM1 🔹	Do not set ISP clock frequency
Clock	125000 👻	AVRISP protocol
Timeout USB	100 🚺	Z USB
Timeout Serial	100	
Default	<u>✓ 0</u> k	X <u>C</u> ancel

Ő az. Aztán elővettem egy dugaszpaneles 2313-t. Mégpedig azért, mert a demó verzióban, ha jól emlékszem 4kByte a határ. Nehogy az okozzon a végén problémát. Menüpont Program -> Send to Chip -> Manual és ismertessük fel vele a Chipet.

2	Elektor / mkII compatible Programmer																	x		
F	File Buffer Chip															-				
] 🚄	/ 🖉 🖻 📙 🚢 📅 📅 🚢 📅 🖿 📭 🖕 🗸																			
Chij	Chip ATtiny2313 Flash size 2 M											ash :	size	2 K	В		EEPROM size	128 Bytes		
Fla	Flash EEPROM Lock and Fuse bits																			
		00	01	02	03	04	05	06	07	08	09	0A	0B	OC	0D	0E	0F			
000)	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF			
010)	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF			
020)	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF			
030)	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	•••••		
040)	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF			
050)	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	•••••		
190	۱T	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF			Ψ.
0	A	Ttin	y23'	13,	FLA	SH	: 20	48,	EPF	ROM	1:1:	28								•
θ	Cł	nip I	D :	1E9	10A															
0	A١	/RI	SPI	MK2	2 det	tecte	ed													-
Filel	FileName FLASH : 2 KB EEPROM : 128 Bytes												11							

Na, ez teljes siker. Most kiolvassuk a Chip tartalmát, mert az ördög nem alszik.

EI EI	ekto	or/	mkI	I co	mpa	atib	le P	rog	ram	me	r								x
File	В	uffe	r	Chi	р														•
	/ 🖉 📴 🚢 📅 📅 🚢 📅 🖿 🗣 🗸																		
Chip		ATti	iny2	231	3					Fla	ash s	size		2 K	B		EEPROM size	128 Bytes	
Flash EEPROM Lock and Fuse bits																			
	00	01	02	03	04	05	06	07	08	09	QΑ	OB	0C	0D	0E	OF		1	*
000	CO	12	95	18	95	18	95	18	95	18	95	18	95	18	95	18	Ŕ.•.•.•.•.•.		
010	95	18	95	18	95	18	95	18	95	18	95	18	95	18	95	18	•.•.•.•.•.•.•.		
020	95	18	95	18	95	18	ED	8F	BF	8D	EB	C8	ΕA	ΕO	2E	4E	•.•.•.iŹżŤëČęŕ.N		
030	27	DD	2E	5D	95	A8	Β7	84	2E	08	7F	87	BF	84	E1	88	'Ý.]•``"⊯ź"á		
040	27	99	BD	81	BD	91	E7	EE	ΕO	FO	E6	AO	ΕO	BO	27	88	""""'çîfđć f"		
050	93	8D	97	31	F7	E9	24	66	EF	7F	BB	71	9A	96	BЗ	82	″Ť−1÷é\$fď∥»qš-ł,		
080	F۵	90	27	89	RR	82	ſF	FR	94	F8	CF	FF	97	31	F7	F1	ä‰ » ĎířřťĎi −1÷ń		Ψ.
🔒 F	LAS	НC	ode	rea	d int	o bu	lfer												
🔒 В	luffe	r rea	bi																
3 A	VRI	SPI	MK2	det!	ecte	ed													Ŧ
FileNa	me						Fl	AS	H : 2	2 KB				EEF	RO	М:	128 Bytes		

Ez is működik. Majd teljes törlés és újra írás.

2	🔣 Elektor / mkII compatible Programmer														x					
F	ile	В	uffe	er (Chij	р														-
4	•	3	7		2	C.		8		Ē	5 1			•						
Chi	P	1	ATt i	iny2	231:	3					Fla	ash s	size		2 K	B		EEPROM size	128 Bytes	
Fla	Flash EEPROM Lock and Fuse bits																			
	0	00	01	02	03	04	05	06	07	08	09	0A	OB	0C	0D	0E	OF			-
00	0 (20	12	95	18	95	18	95	18	95	18	95	18	95	18	95	18	Ŕ.•.•.•.•.•.		
01	0 9	95	18	95	18	95	18	95	18	95	18	95	18	95	18	95	18	•.•.•.•.•.•.•.		
02	0 9	95	18	95	18	95	18	ED	8F	BF	8D	EB	C8	ΕA	ΕO	2E	4E	 폟ŤëČęŕ.N 		
03	0 2	27	DD	2E	5D	95	A8	Β7	84	2E	08	7F	87	BF	84	E1	88	'Ý.]•°',⊯ź"á		
04	0 2	27	99	BD	81	BD	91	E7	EE	ΕO	FO	E6	AO	ΕO	BO	27	88	"""""'çîfđć f*		
05	0 9	33	8D	97	31	F7	E9	24	66	EF	7F	BB	71	9A	96	BЗ	82	″Ť−1÷é\$fď∥»qš-ł,		
06	n le	- 4	90	27	89	RR	82	СF	FR	94	F8	ΓF	FF	97	31	F7	F1	ä‰ » ĎířřřĎi –1≃ń		
θ	12	8ь	ytes	: wri	tten	to F	LAS	Н												
θ	A٧	/RI	SPI	MK2	det?	ecte	ed .													
θ	12	8Ь	ytes	: wri	tten	to F	LAS	Н												-
File	Nan	ne						FI	LAS	H : 2	2 KB				EEF	RO	М :	128 Bytes		11.

És így kijelenthetjük, hogy működik BASCOM alatt is.

A végére hagytam a programozónak egy olyan tulajdonságát, ami már tényleg csak hab a tortán. Ez pedig a "RESCUE_CLOCK". Ami annyit jelent, hogy a PC6(OC.1A/PCINT8) lábon állandó 4MHz található, arra az esetre, ha a céláramkörünkben tanyázó AVR biztosíték bitjeit sikerült úgy beállítani, hogy csak külső órajelre indul. Ilyenkor csatlakozunk az XTAL1 lábra és

ott adjuk neki a jelet. Viszont ügyeljünk arra, hogy ilyen esetben a programozó – vagyis SPI – órajel csak 125kHz legyen, mert csak így biztosítható a stabil működés. Az általam megépített programozón a mérőm 3,97MHz órajelet mutat. Így néz ki a szkópon:



A beállítás: 5V/div és 0.2µS/div. Azért arra vigyázzunk, hogy itt 5V játszik. Egy céláramkörben, ahol esetleg csak 3,3V vagy akár 2,7V található, ott esetleg zokon veszi az AVR. Ha saját magunk fordítjuk a firmware-t, a config.h-ban van egy lehetőség a 4MHz-s jelet átirányítani a PD5(XCK/PCINT12)-s lábra (#define XCK_RESCUE_CLOCK_ENABLE), ami annyit tesz, hogy az órajelet a szintillesztőn keresztül zavarjuk. Akkor viszont megjelenik a PDI és TPI csati lábain. Ezt mélyebben nem tanulmányoztam. Akinek van kedve, az kivesézheti.

Pluto, 2013-06-17