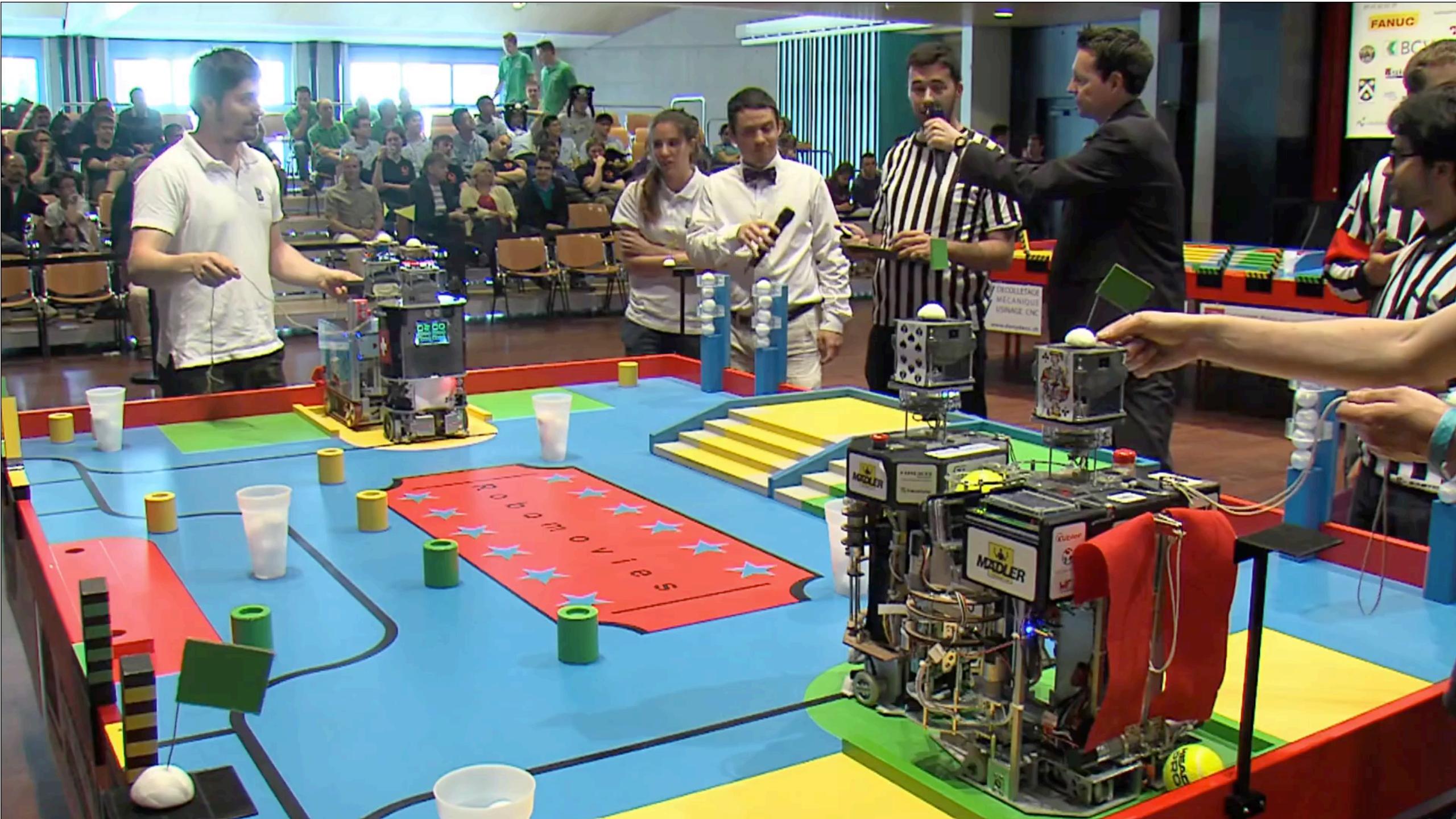
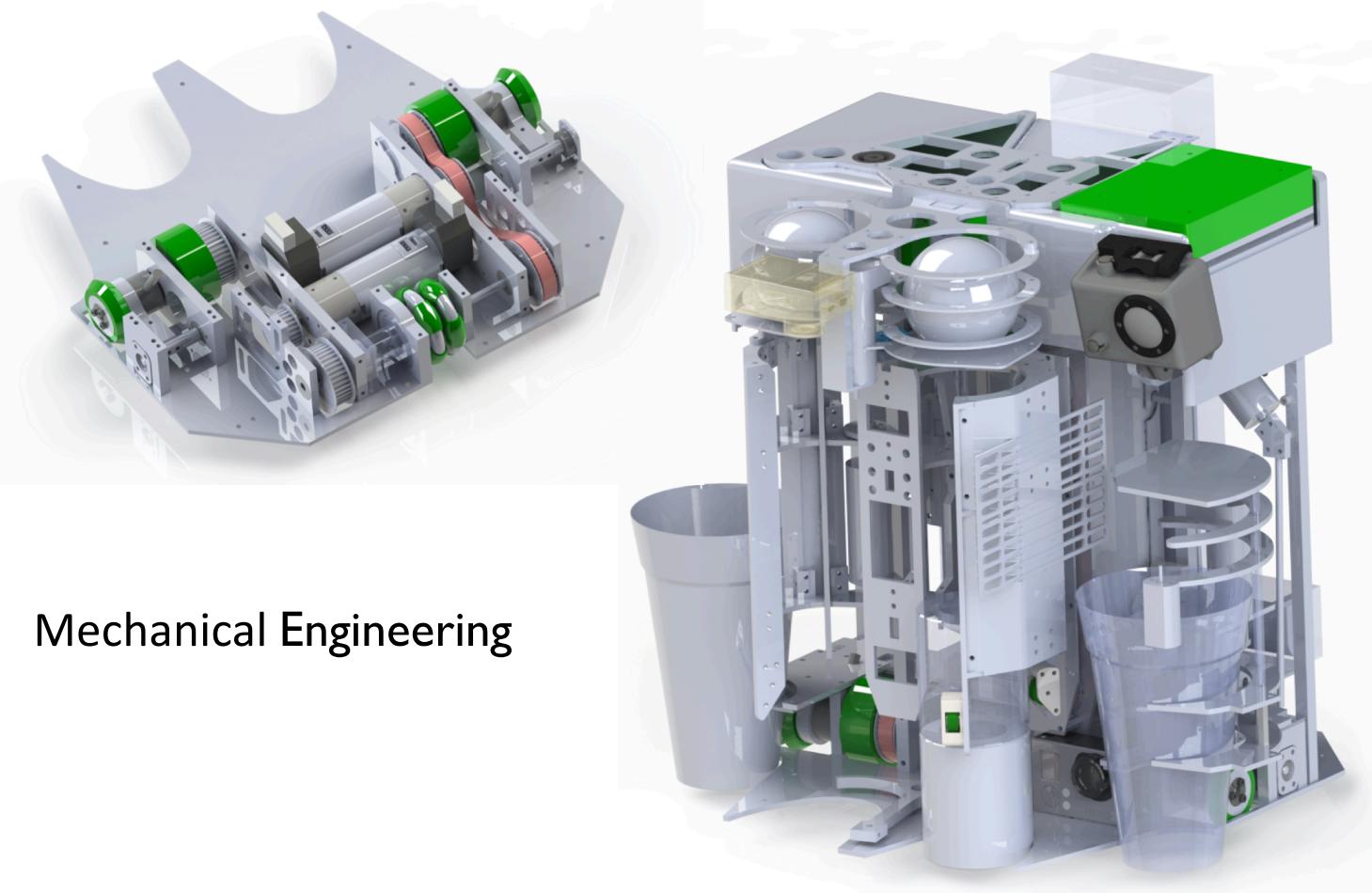
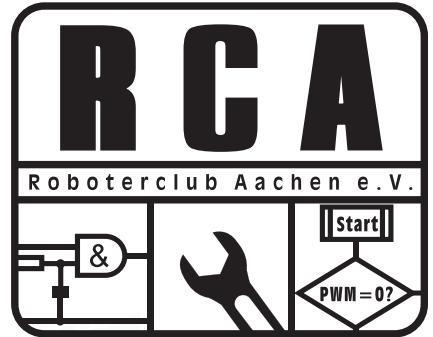


# Data-driven HAL generation

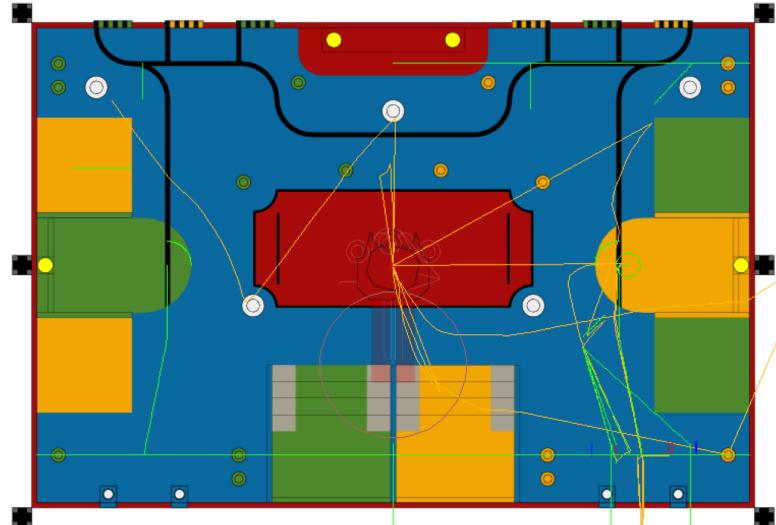
Niklas Hauser

emBO++ 2017

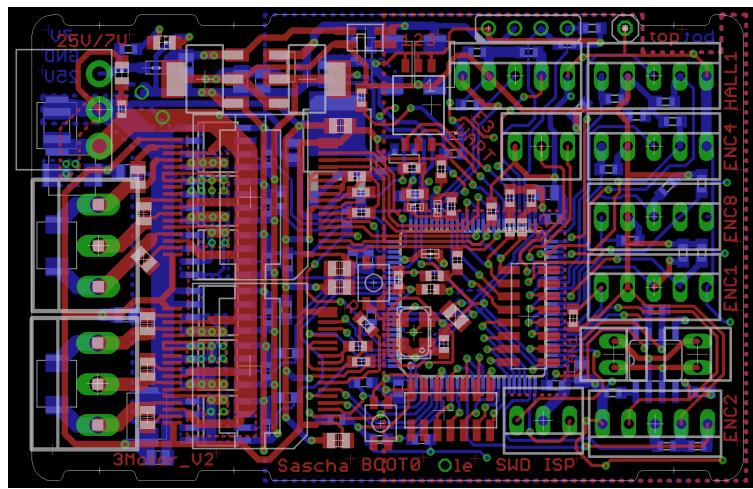




Mechanical Engineering



Software Engineering



Electronic Engineering

**Requirements are different,  
there is no universal HAL!**

	Simulator						
Simulator	AT90CAN	Simulator					
AT90CAN	Mega8	AT90CAN					
Mega8	Mega16	Mega8	Simulator				
Mega16	Mega168	Mega88	Mega88	Simulator			
Mega32	Mega88	Mega32	Xmega128a	STM32F107	Simulator		
<b>Mega168</b>	Tiny44	Mega168	<b>STM32F103</b>	STM32F407	STM32F107	Simulator	
<b>Tiny26</b>	<b>Xmega128a</b>	Xmega128a	<b>STM32F107</b>	<b>STM32F303</b>	STM32F407	STM32F303	
<b>Mega644</b>	<b>Xmega32a</b>	Xmega32a	<b>STM32F407</b>	<b>LPC11C24</b>	STM32F303	STM32F407	
+ FPGA	+ FPGA	+ FPGA	+ FPGA	+ FPGA	+ FPGA	+ FPGA	<b>Tiny85</b>

2009

2010

2011

2012

2013

2014

2015



# “Island ports”

	<b>m328p</b>	<b>f407vg</b>	<b>f303re</b>
<b>core</b>	✓	✓	✓
<b>gpio</b>	✓	✓	✓
<b>clock</b>	✓	✓	✓
<b>uart</b>	✓	✓	✓
<b>spi</b>	✓	✓	✓

The diagram illustrates the presence of SPI ports on three microcontrollers. Blue arrows point downwards from the 'spi' row to the bottom cell of each column, indicating that SPI is supported on all three devices.

f030c6t	f030c8t	f030ccct	f030f4p	f030k6t	f030r8t	f030rcrt	f031c4t	f031c6t	f031e6y	f031f4p	f031f6p	f031g4u	f031g6u	f031k4u	f031k6t	f031k6u	f038c6t	f038e6y	f038f6p	f038g6u	f038k6u	f042c4t
f042c6t	f042c6u	f042f4p	f042f6p	f042g4u	f042g6u	f042k4u	f042k6t	f042k6u	f042t6y	f048c6u	f048g6u	f048t6y	f051c4t	f051c4u	f051c6t	f051c6u	f051c8t	f051c8u	f051k4t	f051k4u	f051k6t	f051k6u
f051k8t	f051k8u	f051r4t	f051r6t	f051r8h	f051r8t	f058c8u	f058r8h	f058r8t	f058t8y	f070c6t	f070cbt	f070f6p	f070rbt	f071cbt	f071cby	f071rbt	f071v8h	f071v8t	f071vhb	f071vbt		
f072c8t	f072c8u	f072cbt	f072cbu	f072cby	f072r8t	f072rbh	f072rbi	f072v8h	f072v8t	f072vhb	f072vbt	f078cbt	f078cby	f078rbh	f078rvt	f078rbt	f078vhb	f078vbt	f091cbt	f091cby	f091ccb	f091cct
f091ccu	f091rbt	f091rch	f091rct	f091rcy	f091vbt	f091vch	f091vct	f098cct	f098ccu	f098rch	f098rct	f098rcy	f098vct	f100e4t	f100c6t	f100c8t	f100c8t	f100r4h	f100r4t	f100r6h	f100r6t	f100r8h
f100r8t	f100rbh	f100rbt	f100rct	f100rdt	f100ret	f100v8t	f100vbt	f100vct	f100vdt	f100vet	f100zct	f100zdt	f100zet	f101c6t	f101c8t	f101cby	f101r4t	f101r6t	f101r8t	f101rbh	f101rbt	
f101rct	f101rdt	f101ret	f101rft	f101rgt	f101t4u	f101t6u	f101t8u	f101tbu	f101v8t	f101vbt	f101vct	f101vdt	f101vet	f101vft	f101zct	f101zdt	f101zet	f101zft	f101zgt	f102c4t	f102c6t	
f102c8t	f102cbt	f102r4t	f102r6t	f102r8t	f102rbt	f103c4t	f103c6t	f103c6u	f103c8t	f103cbt	f103cby	f103r4h	f103r4t	f103r6h	f103r6t	f103r8h	f103r8t	f103rbh	f103rvt	f103rcy	f103rdt	
f103rdy	f103ret	f103rey	f103rft	f103rgt	f103t4u	f103t6u	f103t8u	f103v8h	f103v8t	f103vbh	f103vbi	f103vbt	f103vch	f103vct	f103vdh	f103vdt	f103veh	f103vet	f103vft	f103vgt	f103zch	
f103zct	f103zdh	f103zdt	f103zeh	f103zet	f103zfh	f103zgh	f103zgt	f105r8t	f105rbt	f105rcy	f105v8t	f105vhb	f105vbt	f105vct	f107rbt	f107rct	f107vbt	f107vch	f107vct	f205rbt	f205rcy	
f205ret	f205rey	f205rft	f205rge	f205rgt	f205rgy	f205vbt	f205vct	f205vet	f205vft	f205vgt	f205zct	f205zet	f205zft	f205zgt	f207ich	f207ict	f207ieh	f207iet	f207ifh	f207ift	f207igh	f207igt
f207vct	f207vet	f207vft	f207vgt	f207zct	f207zet	f207zft	f207zgt	f215ret	f215rgt	f215vet	f215vgt	f215zct	f215zgt	f217ieh	f217iet	f217igh	f217igt	f217vet	f217vgt	f217zct	f301c6t	
f301c8t	f301c8y	f301k6u	f301k8u	f301r6t	f301r8t	f302c6t	f302c8t	f302c8y	f302cbt	f302cct	f302k6u	f302k8u	f302r6t	f302r8t	f302rbt	f302ret	f302rdt	f302ret	f302vbt	f302vct	f302vdh	f302vdt
f302veh	f302vet	f302zdt	f302zet	f303c6t	f303c8t	f303cct	f303c6t	f303k6t	f303k8t	f303r6t	f303r8t	f303rbt	f303ret	f303rdt	f303ret	f303vbt	f303vct	f303vcy	f303vdh	f303vdt	f303veh	f303vet
f303vey	f303zdt	f303zet	f318c8t	f318c8y	f318k8u	f328c8t	f334c4t	f334c6t	f334c8t	f334k4t	f334k6t	f334k8t	f334r6t	f334r8t	f358cct	f358rct	f358vct	f373c8t	f373cbt	f373ccct	f373r8t	f373rbt
f373rcy	f373v8h	f373v8t	f373vhb	f373vbt	f373vch	f373vct	f378cct	f378rcy	f378vct	f398vet	f401cbu	f401cby	f401ccu	f401ccy	f401cdy	f401ceu	f401cey	f401r8t	f401rcy	f401rct	f401rdt	
f401ret	f401v8h	f401vbt	f401vch	f401vct	f401vdt	f401vfh	f401vdt	f401veh	f401vet	f405oey	f405ogy	f405rgt	f405vgt	f405zgt	f407ieh	f407iet	f407igh	f407igt	f407vet	f407vgt	f407zct	f410c8u
f410cbu	f410r8t	f410rbt	f410t8y	f410tby	f411ccu	f411ccy	f411ceu	f411cey	f411ret	f411ret	f411vch	f411vct	f411veh	f411vet	f412ceu	f412cgu	f412ret	f412rey	f412rgy	f412veh	f412vet	
f412vgh	f412vgt	f412zej	f412zet	f412zgj	f412zgt	f415ogy	f415rgt	f415vgt	f415zgt	f417ieh	f417iet	f417igh	f417igt	f417vet	f417vgt	f417zct	f417zgt	f427agh	f427aih	f427igh	f427igt	f427iih
f427iit	f427vgt	f427vit	f427zgt	f427zit	f429agh	f429aih	f429bet	f429bgt	f429bit	f429ieh	f429iet	f429igh	f429igt	f429iih	f429iit	f429neh	f429ngh	f429nih	f429vet	f429vgt	f429vit	f429zct
f429zgt	f429zit	f429ziy	f437aih	f437igh	f437igt	f437iih	f437iit	f437vgt	f437vit	f437zgt	f437zit	f439aih	f439bgt	f439bit	f439igh	f439iit	f439iih	f439ngh	f439nih	f439vgt	f439vit	
f439zgt	f439zit	f439ziy	f446mcy	f446mey	f446rct	f446ret	f446vct	f446vet	f446zch	f446zct	f446zeh	f446zej	f446zet	f446zeh	f469agh	f469agy	f469aih	f469aiy	f469bet	f469bgt	f469bit	
f469ieh	f469iet	f469igh	f469igt	f469iit	f469neh	f469ngh	f469nih	f469vet	f469vgt	f469vit	f469zct	f469zgt	f469zit	f479agh	f479iit	f479aih	f479iay	f479vgt	f479bit	f479iht	f479igt	
f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	f479iit	
f746igt	f746neh	f746ngh	f746veh	f746vet	f746vgh	f746vgt	f746zct	f746zgt														
f765iih	f765iit	f765ngh	f765nih	f765vgt	f765vit	f765zgt	f765zit	f767bgt	f767bit	f767igk	f767igt	f767iit	f767iit									
f769bgt	f769bit	f769igt	f769iit	f769iit	f769iit	f769iit	f769iit	f777bit	f777iit													
I011f4p	I011f4u	I011g3u	I011g4u	I011k3t	I011k4t	I011k4u	I021d4p	I021f4p	I021f4u	I021g4u	I021k4t	I021k4u	I031c4t	I031c6t	I031e4y	I031e6y	I031f4p	I031f6p	I031g4u	I031g6u	I031g6us	
I031k4t	I031k4u	I031k6t	I031k6u	I041c4t	I041c6t	I041f6p	I041g6u	I041k6t	I041k6u	I051c6t	I051c8t	I051k6t	I051k6u	I051k8t	I051k8u	I051r6h	I051r6t	I051r8h	I051r8t	I051t6y	I051t8y	I052c6t
I052c8t	I052k6t	I052k6u	I052k8t	I052k8u	I052r6h	I052r6t	I052r8h	I052r8t	I052t6y	I052t8y	I053c6t	I053c8t	I053r6h	I053r6t	I053r8h	I053r8t	I062k8t	I062k8u	I063c8t	I063r8t	I071c8t	I071cbt
I071cby	I071czt	I071czy	I071k8u	I071ktb	I071kbu	I071kzt	I071kzu	I071rbh	I071rbt	I071rzh	I071rzt	I071v8i	I071v8t	I071vbi	I071vbt	I071vzi	I071vzt	I072cby	I072vct	I072vdt	I072czt	I072cby
I072kbt	I072kbu	I072kzt	I072kzu	I072rbh	I072rbi	I072rbz	I072rzi	I072rzt	I072v8i	I072v8t	I072vbi	I072vbt	I072vzi	I073cby	I073cbt	I073czt	I073rbh	I073rzt	I073rbh	I073rzt	I073rzt	I073v8i
I073v8t	I073vbi	I073vbt	I073vzi	I073vzt	I081czt	I081kzt	I081kzu	I082kbt	I082kbu	I082kzt	I082kzu	I083cby	I083cbt	I083czt	I083rbh	I083rzt	I083rbh	I083v8i	I083v8t	I083vbi	I083vbt	I083vzi
I083vzt	I100c6u	I100c6ua	I100r8t	I100r8ta	I100rbt	I100rbta	I100rct	I115c6t	I115c6ta	I115c6u	I115c6ua	I115c8t	I115c8a	I115c8u	I115c8ua	I115c8t	I115c8ta	I115c8u	I115c8ua	I115c8ta	I115c8t	
I1151qdh	I1151qeh	I1151r6h	I1151r6ha	I1151r6t	I1151r8h	I1151r8ha	I1151r8t	I1151rbh	I1151rbha	I1151rbt	I1151rbta	I1151rcy	I1151rcy	I1151rdt	I1151rdt	I1151r8t	I1151r8t	I1151r8t	I1151r8t	I1151r8t	I1151r8t	
I1151v8t	I1151v8ta	I1151v8h	I1151vbh	I1151vbt	I1151vch	I1151vct	I1151vcta															
I1152c8ua	I1152cbt	I1152cbt	I1152cbu	I1152cbua	I1152cct	I1152ccu	I1152cqh															
I1152rcta	I1152rdt	I1152rdt	I1152ret	I1152v8h	I1152v8t	I1152v8a	I1152v8b															
I1162rcta	I1162rdt	I1162rdt	I1162ret	I1162vch	I1162vct	I1162vct	I1162vdt	I1162vdy	I1162vet	I1162vev	I1162vdt											
I1431rby	I1431rci	I1431rci	I1431rcy	I1431vci	I1431vct	I1432kbu	I1432kcu	I1433cby	I1433cby	I1433cct	I1433ccu	I1433ccy	I1433ccy	I1433rbi								
I1443cct	I1443ccu	I1443ccy	I1443rci	I1443rcy	I1443vci	I1443vct	I1471qei	I1471qgi	I1471ret	I1471rgt	I1471vet	I1471vgt	I1471zct	I1471zdt	I1471zgt	I1475rct	I1475ret	I1475vct	I1475vct	I1475vct	I1475vct	I1475vct
I1476jgy	I1476mey	I1476mgy	I1476qei	I1476qgi	I1476ret	I1476ret	I1476vct	I1476vet	I1476vgt	I1476vet	I1476zct	I1476zdt	I1476zgt	I1485jey	I1486qgi	I1486rgt	I1486vgt	I1486vgt	I1486vgt	I1486vgt	I1486vgt	

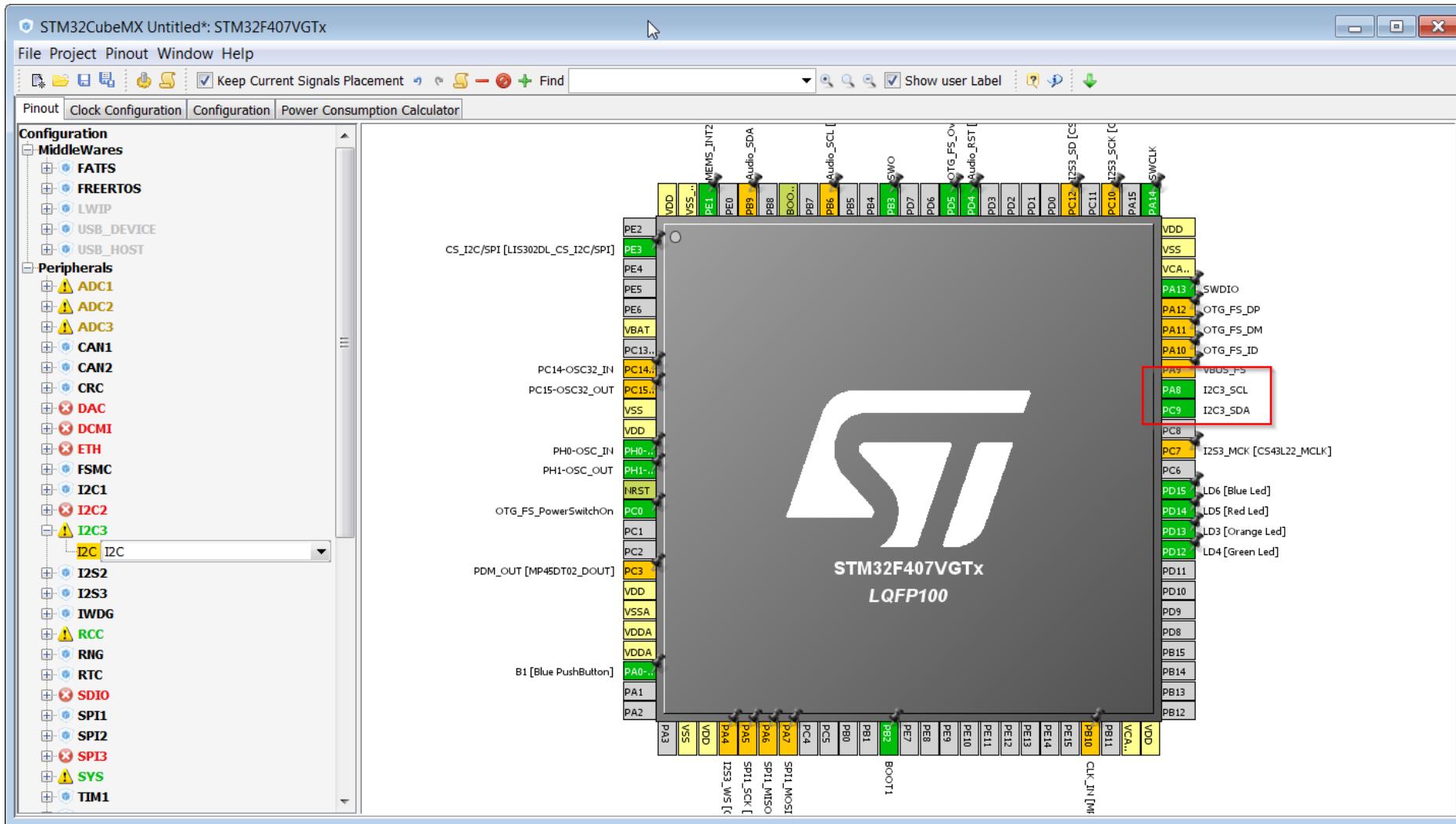
90c128-16ai	90c128-16au	90c128-16mi	90c128-16mu	90c32-16ai	90c32-16au	90c32-16mi	90c32-16mu	90c64-16ai	90c64-16au	90c64-16mi	90c64-16mu	m128-16an	m128-16au	
m128-16mn	m128-16mu	m1280-16au	m1280-16cu	m1280v-8au	m1280v-8cu	m1281-16au	m1281-16mu	m1281v-8au	m1281v-8mu	m1284-au	m1284-mu	m1284-pu	m1284p-an	
m1284p-au	m1284p-mn	m1284p-mu	m1284p-pn	m1284p-pu	m1284p-pu	m128a-aa	m128a-mu	m128l-8an	m128l-8au	m128l-8mn	m128l-8mu	m16-16au	m16-16mu	
m164a-aa	m164a-cu	m164a-mch	m164a-mu	m164a-pu	m164p-20an	m164p-20au	m164p-20mn	m164p-20mu	m164p-20pn	m164p-10pu	m164p-20pu	m164pa-an	m164pa-aa	
m164p-mch	m164pa-mn	m164pa-mu	m164pa-pn	m164pa-pu	m164pv-10an	m164pv-10au	m164pv-10mu	m164pv-10pn	m164pv-10pu	m168-20au	m168-20mu	m168-20pu	m168a-aa	
m168a-ccu	m168a-mmh	m168a-mu	m168a-pu	m168p-20au	m168p-20mu	m168p-20pu	m168pa-an	m168pa-aa	m168pa-ccu	m168pa-mmh	m168pa-mn	m168pa-mu	m168pa-pn	
m168p-pu	m168pb-an	m168pb-aa	m168pb-mn	m168pb-mu	m168pv-10au	m168pv-10mu	m168pv-10pu	m168v-10au	m168v-10mu	m168v-10pu	m169a-an	m169a-aa	m169a-mch	
m169a-mn	m169a-mu	m169p-16au	m169p-16mch	m169p-16mu	m169pa-an	m169pa-aa	m169pa-mch	m169pa-mn	m169pa-mu	m169pv-8au	m169pv-8mch	m169pv-8mu	m16a-aa	
m16a-mu	m16a-pu	m16l-8au	m16l-8mu	m16l-8pu	m16u2-aa	m16u2-mu	m16u4-aa	m16u4-mu	m16u4rc-aa	m16u4rc-mu	m2560-16au	m2560-16cu	m2560v-8au	
m2560v-8cu	m2561-16au	m2561-16mu	m2561v-8au	m2561v-8mu	m32-16au	m32-16mu	m32-16pu	m324a-aa	m324a-cu	m324a-mch	m324a-mu	m324a-pu	m324p-20an	
m324p-20au	m324p-20mn	m324p-20mu	m324p-20pn	m324p-20pu	m324pa-aa	m324pa-cu	m324pa-mch	m324pa-mu	m324pa-pu	m324pb-an	m324pb-aa	m324pb-aur	m324pb-aur	
m324pb-mn	m324pb-mnr	m324pb-mu	m324pb-mur	m324pv-10an	m324pv-10au	m324pv-10mn	m324pv-10mu	m324pv-10pn	m324pv-10pu	m328-aa	m328-mmh	m328-mu	m328-pu	
m328p-an	m328p-aa	m328p-mmh	m328p-mn	m328p-mu	m328p-pn	m328p-pu	m328pb-an	m328pb-aa	m328pb-mn	m328pb-mu	m329-16au	m329-16mu	m329a-an	
m329a-aa	m329a-mn	m329a-mu	m329p-20an	m329p-20au	m329p-20mn	m329p-20mu	m329pa-an	m329pa-aa	m329pa-mn	m329pa-mu	m329pv-10au	m329pv-10mu	m329v-8au	
m329v-8mu	m32a-an	m32a-aa	m32a-mn	m32a-mu	m32a-pu	m32l-8au	m32l-8mu	m32l-8pu	m32u2-aa	m32u2-mu	m32u4-aa	m32u4-mu	m32u4rc-aa	
m32u4rc-mu	m48-20au	m48-20mmh	m48-20muu	m48-20mu	m48-20pu	m48a-aa	m48a-ccu	m48a-mmh	m48a-mu	m48a-pu	m48p-20au	m48p-20muu	m48p-20mu	
m48p-20pu	m48pa-an	m48pa-aa	m48pa-cu	m48pa-mmh	m48pa-mnn	m48pa-mn	m48pa-mu	m48pa-pn	m48pa-pu	m48pb-an	m48pb-aa	m48pb-mn	m48pb-mu	
m48pv-10au	m48pv-10mmu	m48pv-10mu	m48pv-10pu	m48v-10au	m48v-10mmh	m48v-10mmu	m48v-10mu	m48v-10pu	m64-16an	m64-16au	m64-16mn	m64-16mu	m640-16au	
m640-16cu	m640v-8au	m640v-8cu	m644-20au	m644-20mu	m644-20pu	m644a-aa	m644a-mu	m644a-pu	m644p-20an	m644p-20au	m644p-20mn	m644p-20pu	m644p-20pn	
m644p-20pu	m644pa-an	m644pa-aa	m644pa-mn	m644pa-mu	m644pa-pn	m644pa-pu	m644pv-10an	m644pv-10au	m644pv-10mn	m644pv-10mu	m644pv-10pn	m644pv-10pu	m644v-10au	
m644v-10mu	m644v-10pu	m649-16au	m649-16mu	m649a-aa	m649a-mu	m649p-aa	m649p-mu	m649v-8au	m649v-8mu	m64a-an	m64a-aa	m64a-mn	m64a-mu	
m64l-8an	m64l-8au	m64l-8mn	m64l-8mu	m8-16an	m8-16au	m8-16mn	m8-16mu	m8-16pn	m8-16pu	m88-20au	m88-20mu	m88-20pu	m88a-aa	
m88a-ccu	m88a-mmh	m88a-mu	m88a-pu	m88p-20au	m88p-20mu	m88p-20pu	m88pa-an	m88pa-aa	m88pa-ccu	m88pa-mmh	m88pa-mnn	m88pa-mn	m88pa-mu	
m88pa-pn	m88pa-pu	m88pb-aa	m88pb-aa	m88pb-mn	m88pb-mu	m88pv-10au	m88pv-10mu	m88pv-10pu	m88v-10au	m88v-10mu	m88v-10pu	m8a-an	m8a-aa	
m8a-mn	m8a-mu	m8a-pn	m8a-pu	m8l-8an	m8l-8au	m8l-8mn	m8l-8mu	m8l-8pn	m8l-8pu	m8u2-aa	m8u2-mu	t10-mahr	t10-ts8r	
t10-tshr	t102-m7r	t102-m8r	t102-ssfr	t102-ssnr	t102f-m7r	t102f-m8r	t102f-ssfr	t102f-ssnr	t104-ssfr	t104-ssnr	t104f-ssfr	t104f-ssnr	t13-20mmu	
t13-20mu	t13-20pu	t13-20ssu	t13-20su	t13a-mmf	t13a-mmu	t13a-mu	t13a-pu	t13a-sf	t13a-sh	t13a-sn	t13a-ss7	t13a-ssh	t13a-ssu	
t13a-su	t13v-10mmu	t13v-10mu	t13v-10pu	t13v-10ssu	t13v-10su	t1616-mnr	t1616-snr	t1617-mnr	t20-ccu	t20-mmh	t20-ssu	t20-ssu	t20-uur	t20-xu
t24a-ccu	t24a-mf	t24a-mm8	t24a-mmh	t24a-mu	t24a-pu	t24a-ssf	t24a-ssn	t24a-ssu	t25v-10mf	t25v-10mu	t25v-10pu	t25v-10sh	t25v-10sn	
t25v-10ssh	t25v-10ssn	t25v-10ssu	t25v-10su	t25v-20mf	t25v-20mu	t25v-20pu	t25v-20sh	t25v-20sn	t25v-20sh	t25v-20ssn	t25v-20ssu	t25v-20su	t261-20mu	
t261-20pu	t261-20su	t261v-10mu	t261v-10pu	t261v-10su	t4-mahr	t4-ts8r	t4-tshr	t40-mmh	t40-su	t40-xu	t417-mfr	t417-mnr	t441-mmh	
t441-mu	t441-ssu	t44a-ccu	t44a-mf	t44a-mmh	t44a-mu	t44a-pu	t44a-ssf	t44a-ssn	t44a-ssu	t45v-10mu	t45v-10pu	t45v-10sh	t45v-10su	
t45v-10xu	t45v-20mu	t45v-20pu	t45v-20sh	t45v-20su	t45v-20xu	t461-20mu	t461-20pu	t461-20su	t461v-10mu	t461v-10pu	t461v-10su	t48-aa	t48-ccu	
t48-mmh	t48-mmu	t48-mu	t48-pu	t5-mahr	t5-ts8r	t5-tshr	t814-ssfr	t814-ssnr	t814-ssnres	t816-mfr	t816-mnr	t816-mnres	t816-sfr	
t816-snr	t817-mfr	t817-mnr	t817-mnres	t840-xbt	t840-xnr	t841-mmh	t841-mu	t841-ssu	t84a-ccu	t84a-mmh	t84a-mu	t84a-pu	t84a-ssf	
t84a-ssu	t85v-10mu	t85v-10pu	t85v-10sh	t85v-10su	t85v-20mu	t85v-20pu	t85v-20sh	t85v-20su	t861-20mu	t861-20pu	t861-20su	t861v-10mu	t861v-10pu	
t861v-10su	t88-aa	t88-ccu	t88-mmh	t88-mmu	t88-pu	t9-mahr	t9-ts8r	t9-tshr	x128a1-aa	x128a1-cu	x128a1-aa	x128a1-uu	x128a1-c7u	
x128a1u-cu	x128a3-aa	x128a3-mu	x128a3u-aa	x128a3u-mh	x128a4u-aa	x128a4u-cu	x128a4u-mh	x128b1-aa	x128b3-aa	x128b3-mu	x128c3-aa	x128c3-mh	x128d3-aa	
x128d3-mu	x128d4-aa	x128d4-cu	x128d4-mu	x16a4-aa	x16a4-mu	x16a4-aa	x16a4-cu	x16a4-mh	x16c4-aa	x16c4-cu	x16d4-aa	x16d4-mu		
x16e5-aa	x16e5-mu	x192a3-aa	x192a3-mu	x192a3u-aa	x192a3u-mh	x192c3-aa	x192c3-mh	x192d3-aa	x192d3-mu	x256a3-aa	x256a3-mu	x256a3b-aa	x256a3b-mu	
x256a3bu-aa	x256a3bu-mh	x256a3bu-aa	x256a3u-mh	x256c3-aa	x256c3-mh	x256d3-aa	x256d3-mu	x32a4-aa	x32a4-mu	x32a4u-aa	x32a4u-cu	x32a4u-mh	x32c3-aa	
x32c3-mh	x32c4-aa	x32c4-cu	x32c4-mh	x32d3-aa	x32d3-mu	x32d4-aa	x32d4-mu	x32e5-aa	x32e5-mu	x384c3-aa	x384c3-mh	x384d3-aa	x384d3-mh	
x64a1-aa	x64a1-cu	x64a1u-aa	x64a1u-c7u	x64a1u-cu	x64a3-aa	x64a3-mu	x64a3u-aa	x64a3u-mh	x64a4-aa	x64a4u-aa	x64a4u-cu	x64a4u-mh	x64b1-aa	
x64b3-mu	x64c3-aa	x64c3-mh	x64d3-aa	x64d3-mu	x64d4-aa	x64d4-cu	x64d4-mu	x8e5-aa	x8e5-mu					

Arduino  
Riot  
TinyOS  
ChibiOS  
Zephyr  
CMSIS  
mbed  
Contiki  
  
& documentation

# Transfer knowledge and code

1. between devices

2. between HALs



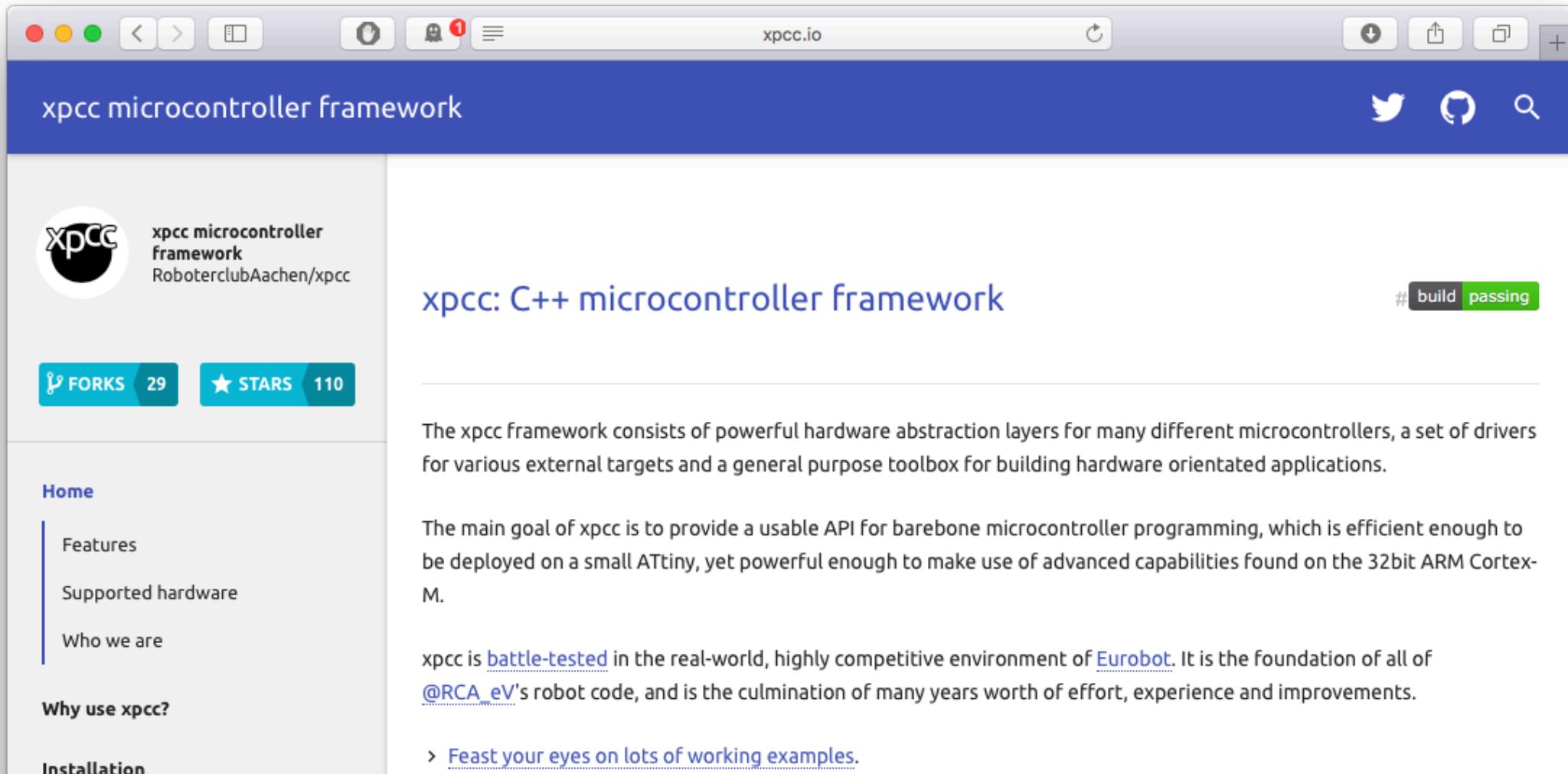
STM32F103R(D-E)Yx.xml	STM32F103R(F-G)Tx.xml	STM32F103T(4-6)Ux.xml	STM32F103T(8-B)Ux.xml	STM32F103V(8-B)Hx.xml	STM32F103V(8-B)Tx.xml	STM32F103V(C-D-E)Hx.xml	STM32F103V(C-D-E)Tx.xml
STM32F103Z(C-D-E)Tx.xml	STM32F103Z(F-G)Hx.xml	STM32F103Z(F-G)Tx.xml	STM32F105R(8-B-C)Tx.xml	STM32F105V(8-B-C)Tx.xml	STM32F105VBHx.xml	STM32F107R(B-C)Tx.xml	STM32F107V(B-C)Tx.xml
STM32F205V...(-F-G)Tx.xml	STM32F205Z(C-E-F-G)Tx.xml	STM32F207I(C-E-F-G)Hx.xml	STM32F207I(C-E-F-G)Tx.xml	STM32F207V(C-E-F-G)Tx.xml	STM32F207Z(C-E-F-G)Tx.xml	STM32F215R(E-G)Tx.xml	STM32F215V(E-G)Tx.xml
STM32F217V(E-G)Tx.xml	STM32F217Z(E-G)Tx.xml	STM32F301C(6-8)Tx.xml	STM32F301C8Yx.xml	STM32F301K(6-8)Ux.xml	STM32F301R(6-8)Tx.xml	STM32F302C(6-8)Tx.xml	STM32F302C(B-C)Tx.xml
STM32F302R(B-C)Tx.xml	STM32F302R(D-E)Tx.xml	STM32F302V(B-C)Tx.xml	STM32F302V(D-E)Tx.xml	STM32F302Z(D-E)Tx.xml	STM32F303C(6-8)Tx.xml	STM32F303C(B-C)Tx.xml	STM32F303K(6-8)Tx.xml
STM32F303V(B-C)Tx.xml	STM32F303V(D-E)Tx.xml	STM32F303VEHx.xml	STM32F303Z(D-E)Tx.xml	STM32F318C8Tx.xml	STM32F318C8Yx.xml	STM32F318K8Ux.xml	STM32F328C8Tx.xml
STM32F358CCTx.xml	STM32F358RCTx.xml	STM32F358VCTx.xml	STM32F373C(8-B-C)Tx.xml	STM32F373R(8-B-C)Tx.xml	STM32F373V(8-B-C)Hx.xml	STM32F373V(8-B-C)Tx.xml	STM32F378CCTx.xml
STM32F398VETx.xml	STM32F401C(B-C)Ux.xml	STM32F401C(B-C)Yx.xml	STM32F401C(D-E)Ux.xml	STM32F401C(D-E)Yx.xml	STM32F401R(B-C)Tx.xml	STM32F401R(D-E)Tx.xml	STM32F401V(B-C)Hx.xml
STM32F405Q(E-G)Yx.xml	STM32F405RCTx.xml	STM32F405VGTx.xml	STM32F405ZGTx.xml	STM32F407I(E-G)Hx.xml	STM32F407I(E-G)Tx.xml	STM32F407V(E-G)Tx.xml	STM32F407Z(E-G)Tx.xml
STM32F411C(C-E)Yx.xml	STM32F411CEUx.xml	STM32F411R(C-E)Tx.xml	STM32F411V(C-E)Hx.xml	STM32F411V(C-E)Tx.xml	STM32F415OGYx.xml	STM32F415RGTx.xml	STM32F415VGTx.xml
STM32F417V(E-G)Tx.xml	STM32F417Z(E-G)Tx.xml	STM32F427A(G-I)Hx.xml	STM32F427I(G-I)Hx.xml	STM32F427I(G-I)Tx.xml	STM32F427V(G-I)Tx.xml	STM32F427Z(G-I)Tx.xml	STM32F429A(G-I)Hx.xml
STM32F429N(E-G-I)Hx.xml	STM32F429V(E-G-I)Tx.xml	STM32F429Z(E-G-I)Tx.xml	STM32F429ZIYx.xml	STM32F437AIHx.xml	STM32F437I(G-I)Hx.xml	STM32F437V(G-I)Tx.xml	STM32F437V(G-I)Tx.xml
STM32F439I(G-I)Hx.xml	STM32F439I(G-I)Tx.xml	STM32F439N(G-I)Hx.xml	STM32F439V(G-I)Tx.xml	STM32F439Z(G-I)Tx.xml	STM32F446M(C-E)Yx.xml	STM32F446R(C-E)Tx.xml	STM32F446R(C-E)Tx.xml
STM32F469A(E-G-I)Hx.xml	STM32F469A(E-G-I)Yx.xml	STM32F469B(E-G-I)Tx.xml	STM32F469I(E-G-I)Tx.xml	STM32F469I(G-I)Hx.xml	STM32F469N(E-G-I)Hx.xml	STM32F479A(G-I)Hx.xml	STM32F479A(G-I)Yx.xml
STM32F479N(G-I)Hx.xml	STM32F475I(E-G)Kx.xml	STM32F475I(E-G)Tx.xml	STM32F475V(E-G)Tx.xml	STM32F475Z(E-G)Tx.xml	STM32F476B(E-G)Tx.xml	STM32F476I(E-G)Kx.xml	STM32F476I(E-G)Tx.xml
STM32F746Z(E-G)Yx.xml	STM32F756BGTx.xml	STM32F756IGKx.xml	STM32F756IGTx.xml	STM32F756NGHx.xml	STM32F756VGTx.xml	STM32F756ZGTx.xml	STM32F756ZGYx.xml
STM32L031G(4-6)Ux.xml	STM32L031K(4-6)Tx.xml	STM32L031K(4-6)Ux.xml	STM32L041C(4-6)Tx.xml	STM32L041F6Px.xml	STM32L041G6Ux.xml	STM32L041K6Ux.xml	STM32L051C(6-8)Tx.xml
STM32L051R(6-8)Tx.xml	STM32L051T(6-8)Yx.xml	STM32L052C(6-8)Tx.xml	STM32L052K(6-8)Tx.xml	STM32L052K(6-8)Ux.xml	STM32L052R(6-8)Hx.xml	STM32L052R(6-8)Tx.xml	STM32L052T(6-8)Yx.xml
STM32L062K8Tx.xml	STM32L062K8Ux.xml	STM32L063C8Tx.xml	STM32L063R8Tx.xml	STM32L071C(B-Z)Tx.xml	STM32L071C(B-Z)Yx.xml	STM32L071C8Tx.xml	STM32L071K(B-Z)Tx.xml
STM32L071R(B-Z)Tx.xml	STM32L071V(B-Z)Ix.xml	STM32L071V(B-Z)Tx.xml	STM32L071V8Ix.xml	STM32L071V8Tx.xml	STM32L072C(B-Z)Tx.xml	STM32L072C(B-Z)Yx.xml	STM32L072K(B-Z)Tx.xml
STM32L072R(B-Z)Tx.xml	STM32L072V(B-Z)Ix.xml	STM32L072V(B-Z)Tx.xml	STM32L072V8Ix.xml	STM32L072V8Tx.xml	STM32L073C(B-Z)Tx.xml	STM32L073R(B-Z)Hx.xml	STM32L073R(B-Z)Tx.xml
STM32L073V8Tx.xml	STM32L081CZTx.xml	STM32L081KZTx.xml	STM32L082K(B-Z)Tx.xml	STM32L082K(B-Z)Ux.xml	STM32L083C(B-Z)Tx.xml	STM32L083R(B-Z)Hx.xml	STM32L083R(B-Z)Tx.xml
STM32L083V8Tx.xml	STM32L100C6Ux.xml	STM32L100C6UxA.xml	STM32L100R(8-B)Tx.xml	STM32L100R(8-B)TxA.xml	STM32L100RCTx.xml	STM32L151C(6-8-B)Tx.xml	STM32L151C(6-8-B)TxA.xml
STM32L151CCUx.xml	STM32L151QCHx.xml	STM32L151QDHx.xml	STM32L151QEHx.xml	STM32L151R(6-8-B)Hx.xml	STM32L151R(6-8-B)HxA.xml	STM32L151R(6-8-B)Tx.xml	STM32L151R(6-8-B)TxA.xml
STM32L151RDTx.xml	STM32L151RDYx.xml	STM32L151RETx.xml	STM32L151UCYx.xml	STM32L151V(8-B)Hx.xml	STM32L151V(8-B)HxA.xml	STM32L151V(8-B)Tx.xml	STM32L151V(8-B)TxA.xml
STM32L151VDTx.xml	STM32L151VDYx.xml	STM32L151VETx.xml	STM32L151VEYx.xml	STM32L151ZCTx.xml	STM32L151ZDTx.xml	STM32L151ZETx.xml	STM32L152C(6-8-B)Tx.xml
STM32L152CCTx.xml	STM32L152CCUx.xml	STM32L152QCHx.xml	STM32L152QDHx.xml	STM32L152QEHx.xml	STM32L152R(6-8-B)Hx.xml	STM32L152R(6-8-B)HxA.xml	STM32L152R(6-8-B)Tx.xml
STM32L152RDTx.xml	STM32L152RDYx.xml	STM32L152RETx.xml	STM32L152V(8-B)Hx.xml	STM32L152V(8-B)HxA.xml	STM32L152V(8-B)Tx.xml	STM32L152V(8-B)TxA.xml	STM32L152VCHx.xml
STM32L152VDTx.xml	STM32L152VETx.xml	STM32L152VEYx.xml	STM32L152ZCTx.xml	STM32L152ZDTx.xml	STM32L152ZETx.xml	STM32L162QDHx.xml	STM32L162RCTx.xml
STM32L162RETx.xml	STM32L162VCHx.xml	STM32L162VCTx.xml	STM32L162VCTxA.xml	STM32L162VDTx.xml	STM32L162VDYx.xml	STM32L162VETx.xml	STM32L162VEYx.xml
STM32L471Q(E-G)Ix.xml	STM32L471R(E-G)Tx.xml	STM32L471V(E-G)Tx.xml	STM32L471Z(E-G)Tx.xml	STM32L475J(E-G)Yx.xml	STM32L475Q(E-G)Ix.xml	STM32L475R(C-E-G)Tx.xml	STM32L475V(C-E-G)Tx.xml
STM32L476Q(E-G)Ix.xml	STM32L476R(C-E-G)Tx.xml	STM32L476V(C-E-G)Tx.xml	STM32L476Z(E-G)Tx.xml	STM32L485J(C-E)Yx.xml	STM32L486QGlx.xml	STM32L486RGTx.xml	

STM32F103V(8-B)Hx.xml STM32F103V(8-B)Tx.xml STM32F103V(C-D-E)Hx.xml STM32F103V(C-D-E)Tx.xml  
STM32F105V(8-B-C)Tx.xml STM32F105VBHx.xml STM32F107R(B-C)Tx.xml STM32F107V(B-C)Tx.xml  
STM32F207V(C-E-F-G)Tx.xml STM32F207Z(C-E-F-G)Tx.xml STM32F215R(E-G)Tx.xml STM32F215V(E-G)Tx.xml  
STM32F301K(6-8)Ux.xml STM32F301R(6-8)Tx.xml STM32F302C(6-8)Tx.xml STM32F302C(B-C)Tx.xml  
STM32F302Z(D-E)Tx.xml STM32F303C(6-8)Tx.xml STM32F303C(B-C)Tx.xml STM32F303K(6-8)Tx.xml  
STM32F318C8Tx.xml STM32F318C8Yx.xml STM32F318K8Ux.xml STM32F328C8Tx.xml  
STM32F373R(8-B-C)Tx.xml STM32F373V(8-B-C)Hx.xml STM32F373V(8-B-C)Tx.xml STM32F378CCTx.xml  
STM32F401C(D-E)Yx.xml STM32F401R(B-C)Tx.xml STM32F401R(D-E)Tx.xml STM32F401V(B-C)Hx.xml  
STM32F407(E-G)Hx.xml STM32F407(E-G)Tx.xml STM32F407V(E-G)Tx.xml STM32F407Z(E-G)Tx.xml  
STM32F411V(C-E)Tx.xml STM32F415OGYx.xml STM32F415RGTx.xml STM32F415VGTx.xml  
STM32F427I(G-I)Tx.xml STM32F427V(G-I)Tx.xml STM32F427Z(G-I)Tx.xml STM32F429A(G-I)Hx.xml  
STM32F437AIHx.xml STM32F437I(G-I)Hx.xml STM32F437I(G-I)Tx.xml STM32F437V(G-I)Tx.xml  
STM32F439Z(G-I)Tx.xml STM32F439ZIYx.xml STM32F446M(C-E)Yx.xml STM32F446R(C-E)Tx.xml  
STM32F469I(G-I)Hx.xml STM32F469N(E-G-I)Hx.xml STM32F479A(G-I)Hx.xml STM32F479A(G-I)Yx.xml  
STM32F745Z(E-G)Tx.xml STM32F746B(E-G)Tx.xml STM32F746I(E-G)Kx.xml STM32F746I(E-G)Tx.xml  
STM32F756NGHx.xml STM32F756VGTx.xml STM32F756ZGTx.xml STM32F756ZGYx.xml  
STM32L041F6Px.xml STM32L041G6Ux.xml STM32L041K6Ux.xml STM32L051C(6-8)Tx.xml  
STM32L052K(6-8)Ux.xml STM32L052R(6-8)Hx.xml STM32L052R(6-8)Tx.xml STM32L052T(6-8)Yx.xml  
STM32L071C(B-Z)Tx.xml STM32L071C(B-Z)Yx.xml STM32L071C8Tx.xml STM32L071K(B-Z)Tx.xml  
STM32L071V8Tx.xml STM32L072C(B-Z)Tx.xml STM32L072C(B-Z)Yx.xml STM32L072K(B-Z)Tx.xml  
STM32L072V8Tx.xml STM32L073C(B-Z)Tx.xml STM32L073R(B-Z)Hx.xml STM32L073R(B-Z)Tx.xml  
STM32L082K(B-Z)Ux.xml STM32L083C(B-Z)Tx.xml STM32L083R(B-Z)Hx.xml STM32L083R(B-Z)Tx.xml  
STM32L100R(8-B)Tx.A.xml STM32L100RCTx.xml STM32L151C(6-8-B)Tx.xml STM32L151C(6-8-B)Tx.A.xml  
STM32L151R(6-8-B)Hx.xml STM32L151R(6-8-B)HxA.xml STM32L151R(6-8-B)Tx.xml STM32L151R(6-8-B)Tx.A.xml  
STM32L151V(8-B)Hx.xml STM32L151V(8-B)HxA.xml STM32L151V(8-B)Tx.xml STM32L151V(8-B)Tx.A.xml  
STM32L151ZCTx.xml STM32L151ZDTx.xml STM32L151ZETx.xml STM32L152C(6-8-B)Tx.xml  
STM32L152QEHx.xml STM32L152R(6-8-B)Hx.xml STM32L152R(6-8-B)HxA.xml STM32L152R(6-8-B)Tx.xml  
STM32L152V(8-B)HxA.xml STM32L152V(8-B)Tx.xml STM32L152V(8-B)Tx.A.xml STM32L152VCHx.xml  
STM32L152ZDTx.xml STM32L152ZETx.xml STM32L162QDHx.xml STM32L162RCTx.xml  
STM32L162VDTx.xml STM32L162VDYxX.xml STM32L162VETx.xml STM32L162VEYx.xml  
STM32L475J(E-G)Yx.xml STM32L475Q(E-G)Ix.xml STM32L475R(C-E-G)Tx.xml STM32L475V(C-E-G)Tx.xml  
STM32L485J(C-E)Yx.xml STM32L486JGYx.xml STM32L486QGlx.xml STM32L486RGTx.xml



# Device Files

# Device File → SCons + Jinja2 → HAL



The screenshot shows a GitHub repository page for the xpcc microcontroller framework. The header bar includes standard browser controls, a search bar with the text "xpcc.io", and social sharing icons for Twitter, GitHub, and a magnifying glass.

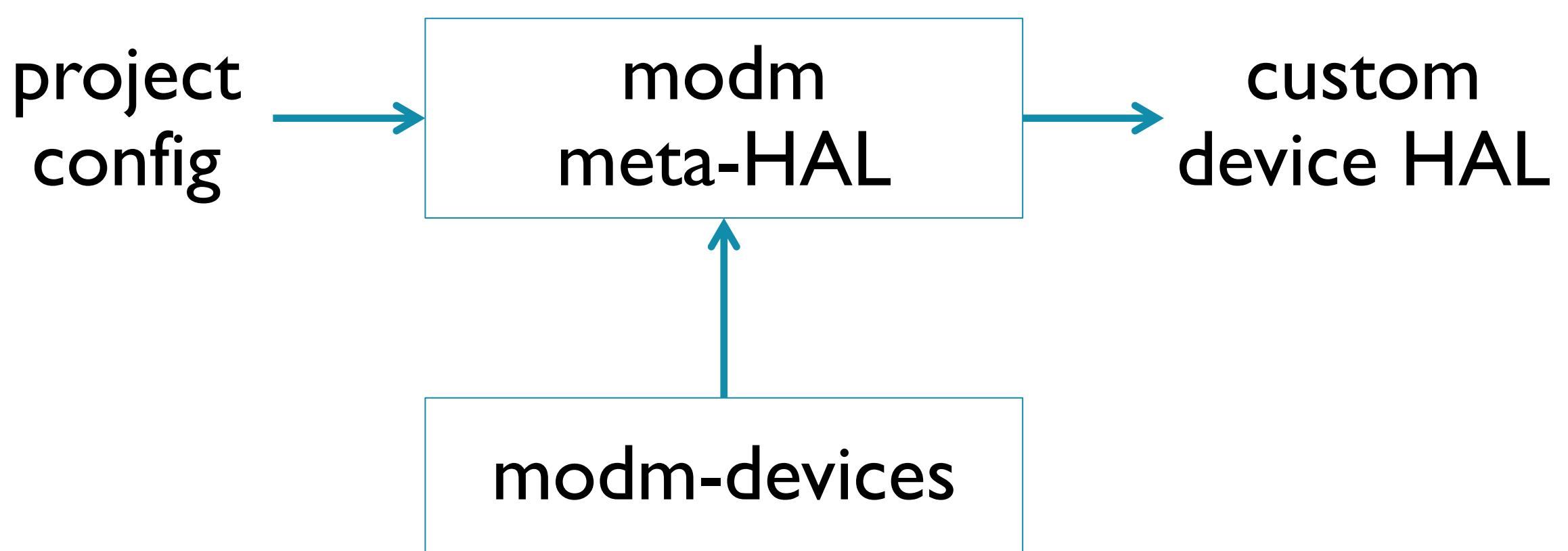
The main content area has a blue header with the text "xpcc microcontroller framework". On the left, there's a sidebar with the xpcc logo, the repository name "xpcc microcontroller framework", and developer information "RoboterclubAachen/xpcc". Below this are buttons for "FORKS 29" and "STARS 110".

The main content area features a large heading "xpcc: C++ microcontroller framework" and a green button "# build passing".

The page describes the xpcc framework as consisting of powerful hardware abstraction layers for many different microcontrollers, a set of drivers for various external targets, and a general purpose toolbox for building hardware orientated applications. It highlights the main goal of providing a usable API for barebone microcontroller programming, suitable for both small ATtiny and advanced 32bit ARM Cortex-M targets.

It also mentions that xpcc is "battle-tested" in the real-world, highly competitive environment of Eurobot, and is the foundation of all of @RCA\_eV's robot code. The page concludes with a call to action: "Feast your eyes on lots of working examples."

# modm is our next generation **toolbox**



project  
config

modm  
meta-HAL

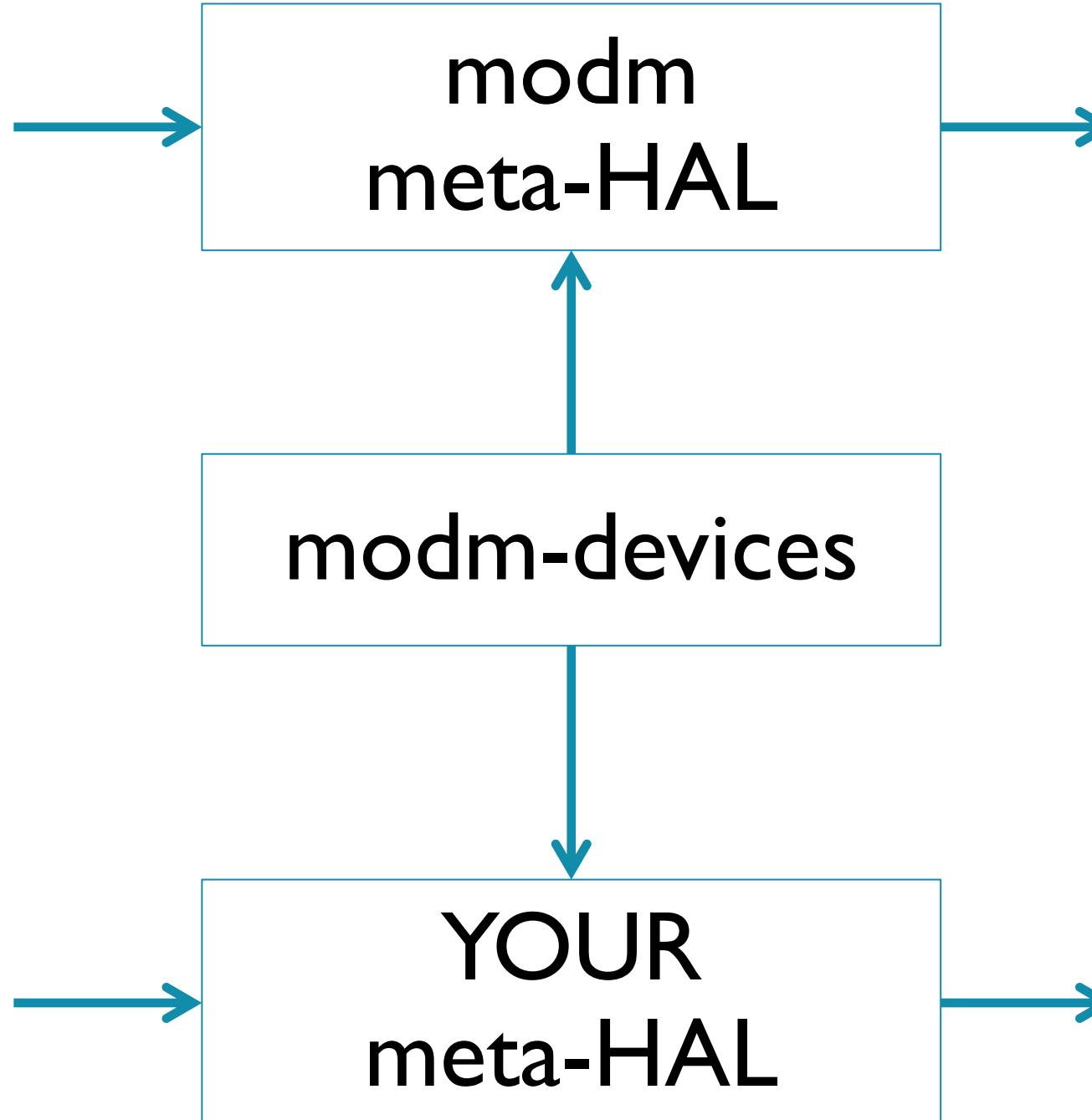
custom  
device HAL

modm-devices

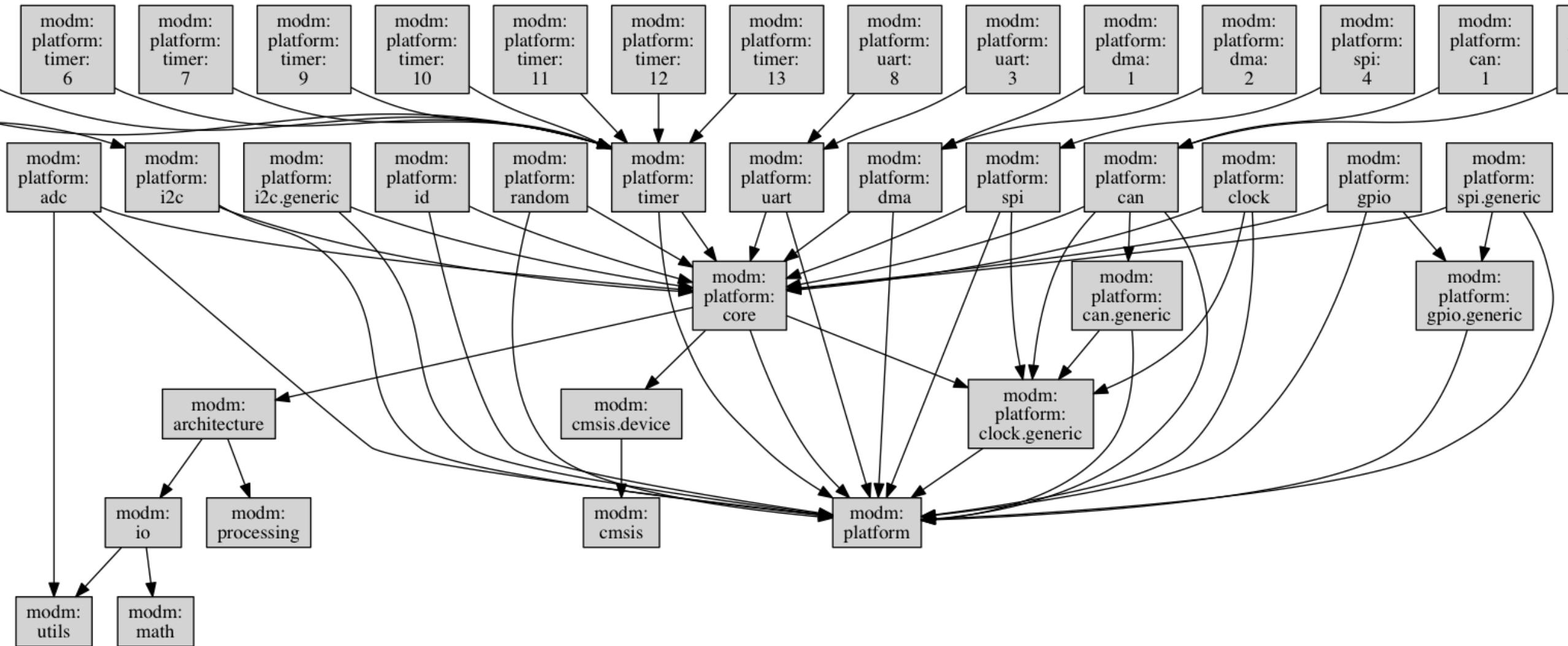
YOUR  
config

YOUR  
meta-HAL

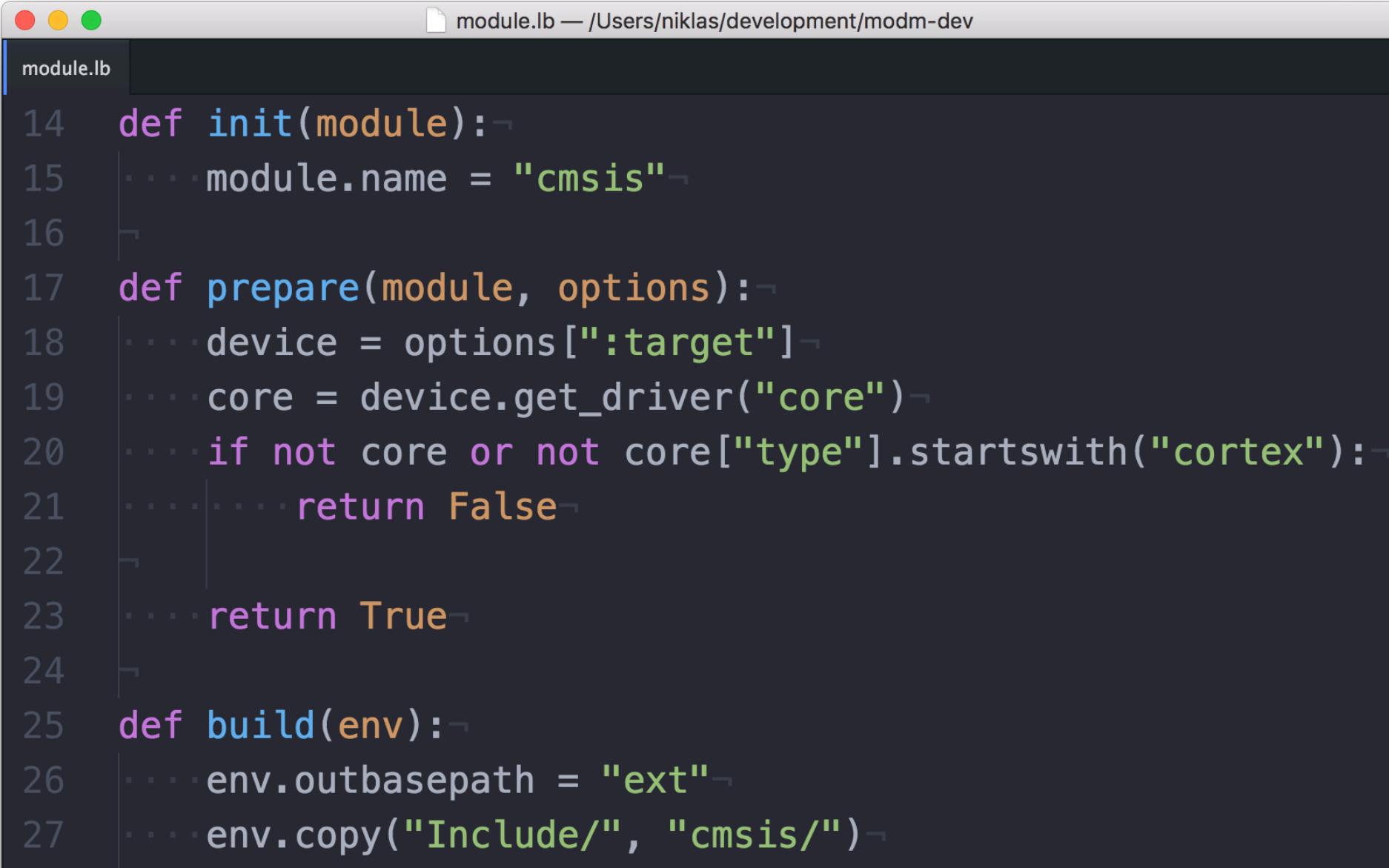
YOUR  
device HAL



# modularize your HAL with lbuild



# Ibuild modules are Python



A screenshot of a terminal window titled "module.lb — /Users/niklas/development/modm-dev". The window contains the following Python code:

```
module.lb
14 def init(module):
15     module.name = "cmsis"
16
17 def prepare(module, options):
18     device = options[:target]
19     core = device.get_driver("core")
20     if not core or not core["type"].startswith("cortex"):
21         return False
22
23     return True
24
25 def build(env):
26     env.outbasepath = "ext"
27     env.copy("Include/", "cmsis/")
```

# Ibuild modules have options

```
1. niklas@stockholm: ~/development/modm-dev/modm-sandbox/test (zsh)
modm:platform:can:2:buffer.rx = 32 [1 ... 65534]
modm:platform:can:2:buffer.tx = 32 [1 ... 65534]
modm:platform:core:allocator = newlib [block_allocator, newlib, tlsf]

Dynamic memory allocation strategy

modm:platform:core:hardfault.log = disabled [basic, disabled, full]
modm:platform:core:hardfault.log uart = 1 [1, 2, 3, 4, 5, 6, 7, 8]
modm:platform:core:main_stack_size = 3040 [256 ... 65536]

Minimum size of the application main stack

modm:platform:core:vector_table_location = fastest [fastest, ram, rom]

Vector table location in ROM or RAM

modm:platform:gpio:enable_gpio = True [True, False]
```

# Ibuild options are documented

```
t" discover-option --option=":::vector_table_location"
modm:platform:core:vector_table_location
=====
Current value: fastest
Possible values: fastest, ram, rom

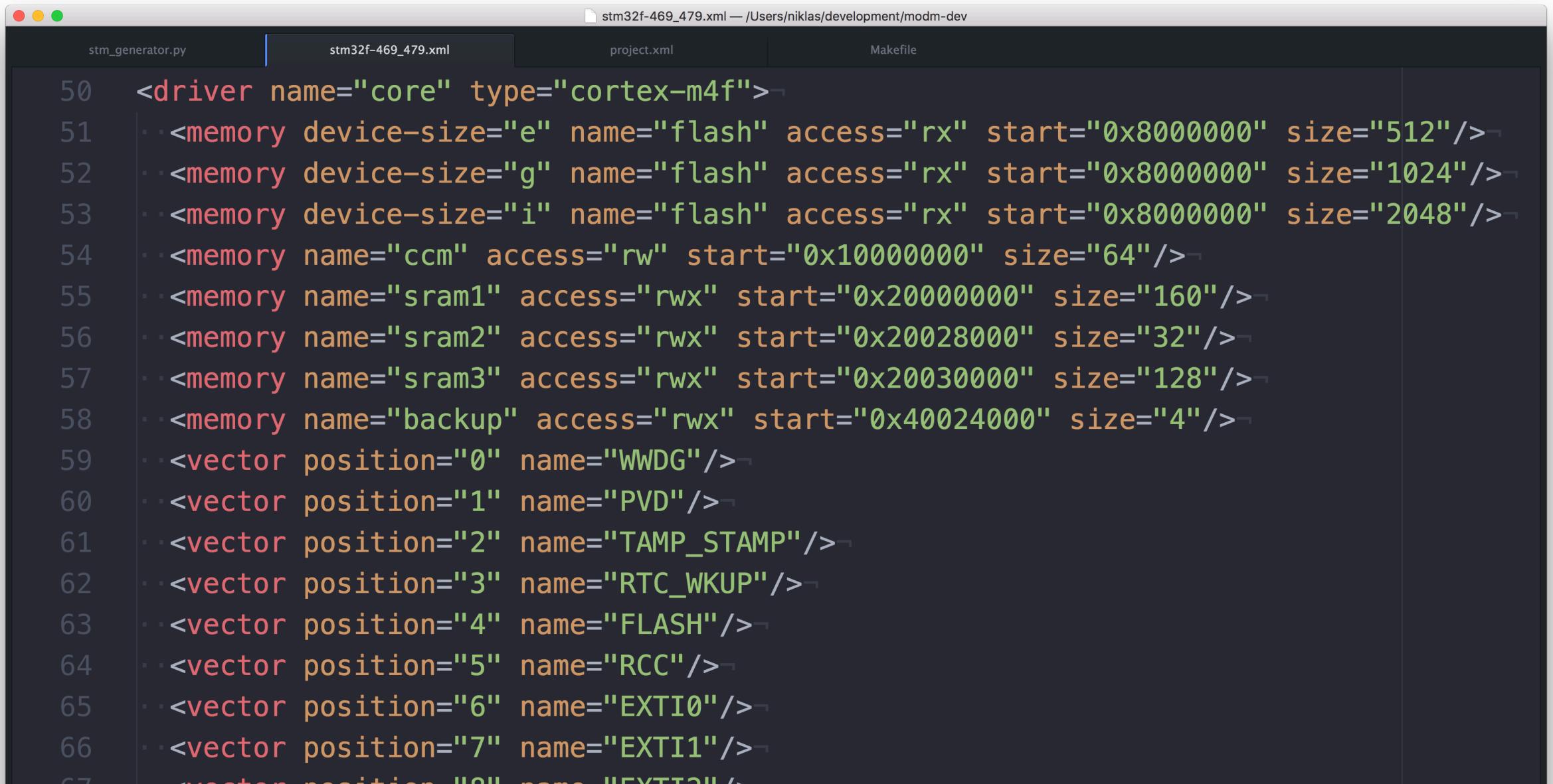
Vector table location in ROM or RAM

The vector table is always stored in ROM and copied to RAM by the startup script
if required. You can modify the RAM vector table using the CMSIS NVIC functions:

- `void NVIC_SetVector(IRQn_Type IRQn, uint32_t vector)`
- `uint32_t NVIC_GetVector(IRQn_Type IRQn)`

For applications that do not modify the vector table at runtime, relocation to
RAM is not necessary and can save a few hundred bytes of static memory.
```

# modm:platform:core for Cortex-M



The screenshot shows a code editor window with a dark theme. The title bar reads "stm32f-469\_479.xml — /Users/niklas/development/modm-dev". The tabs at the top are "stm\_generator.py", "stm32f-469\_479.xml" (which is active), "project.xml", and "Makefile". The main content area displays the following XML code:

```
50 <driver name="core" type="cortex-m4f">
51   <memory device-size="e" name="flash" access="rx" start="0x8000000" size="512"/>
52   <memory device-size="g" name="flash" access="rx" start="0x8000000" size="1024"/>
53   <memory device-size="i" name="flash" access="rx" start="0x8000000" size="2048"/>
54   <memory name="ccm" access="rw" start="0x10000000" size="64"/>
55   <memory name="sram1" access="rwx" start="0x20000000" size="160"/>
56   <memory name="sram2" access="rwx" start="0x20028000" size="32"/>
57   <memory name="sram3" access="rwx" start="0x20030000" size="128"/>
58   <memory name="backup" access="rwx" start="0x40024000" size="4"/>
59   <vector position="0" name="WWDG"/>
60   <vector position="1" name="PVD"/>
61   <vector position="2" name="TAMP_STAMP"/>
62   <vector position="3" name="RTC_WKUP"/>
63   <vector position="4" name="FLASH"/>
64   <vector position="5" name="RCC"/>
65   <vector position="6" name="EXTI0"/>
66   <vector position="7" name="EXTI1"/>
67   <vector position="8" name="EXTI2"/>
```

# AVR memory maps are your foes

```
? ADC:RB:MUX>
[INFO] dfg.memory_query: 1 matches found for ADC -> RB -> MUX:
attiny[4|5|9|10|11|12|13|15|20|24|25|26|40|43|44|45|48|80|84|85|87|88|102|104|167|261|417|441|461|814|816|817|828|840|841|86
[|a|u]
ADC:ADCSRB           8-bit          !at[4|9|11|12|80|840|2313|4313]
!at[4|9|11|12|15|26|80|417|814|816|817|840|1616|1617|2313|4313]:
    7             6             5             4             3             2 (0)         1 (0)         0 (0)
    |             -             |             -             |             -             |             -             |             ADTS2         |             ADTS1         |             ADTS0
    |             -             |             -             |             -             |             -             |             -             |             ADTS2         |             ADTS1         |             ADTS0
at[24|25|44|45|84|85|87|167|261|461|861]:
    7 (0)         6             5             4             3             2 (0)         1 (0)         0 (0)
    |             BIN           |             -             |             -             |             -             |             -             |             ADTS2         |             ADTS1         |             ADTS0
    |             -             |             -             |             -             |             -             |             -             |             ADTS2         |             ADTS1         |             ADTS0
at[25|45|85|261|461|861]:
    7 (0)         6             5 (0)         4             3             2 (0)         1 (0)         0 (0)
    |             BIN           |             -             |             IPR           |             -             |             -             |             ADTS2         |             ADTS1         |             ADTS0
    |             -             |             -             |             IPR           |             -             |             -             |             ADTS2         |             ADTS1         |             ADTS0
at[24|43|44|84]:
    7             6             5             4 (0)         3             2 (0)         1 (0)         0 (0)
    |             -             |             -             |             ADLAR         |             -             |             ADTS2         |             ADTS1         |             ADTS0
    |             -             |             -             |             ADLAR         |             -             |             ADTS2         |             ADTS1         |             ADTS0
at[261|461|861]:
    7 (0)         6 (0)         5 (0)         4 (0)         3 (0)         2 (0)         1 (0)         0 (0)
    |             BIN           |             GSEL          |             IPR           |             REFS2         |             MUX5          |             ADTS2         |             ADTS1         |             ADTS0
```

# STM32 memory maps are your friends

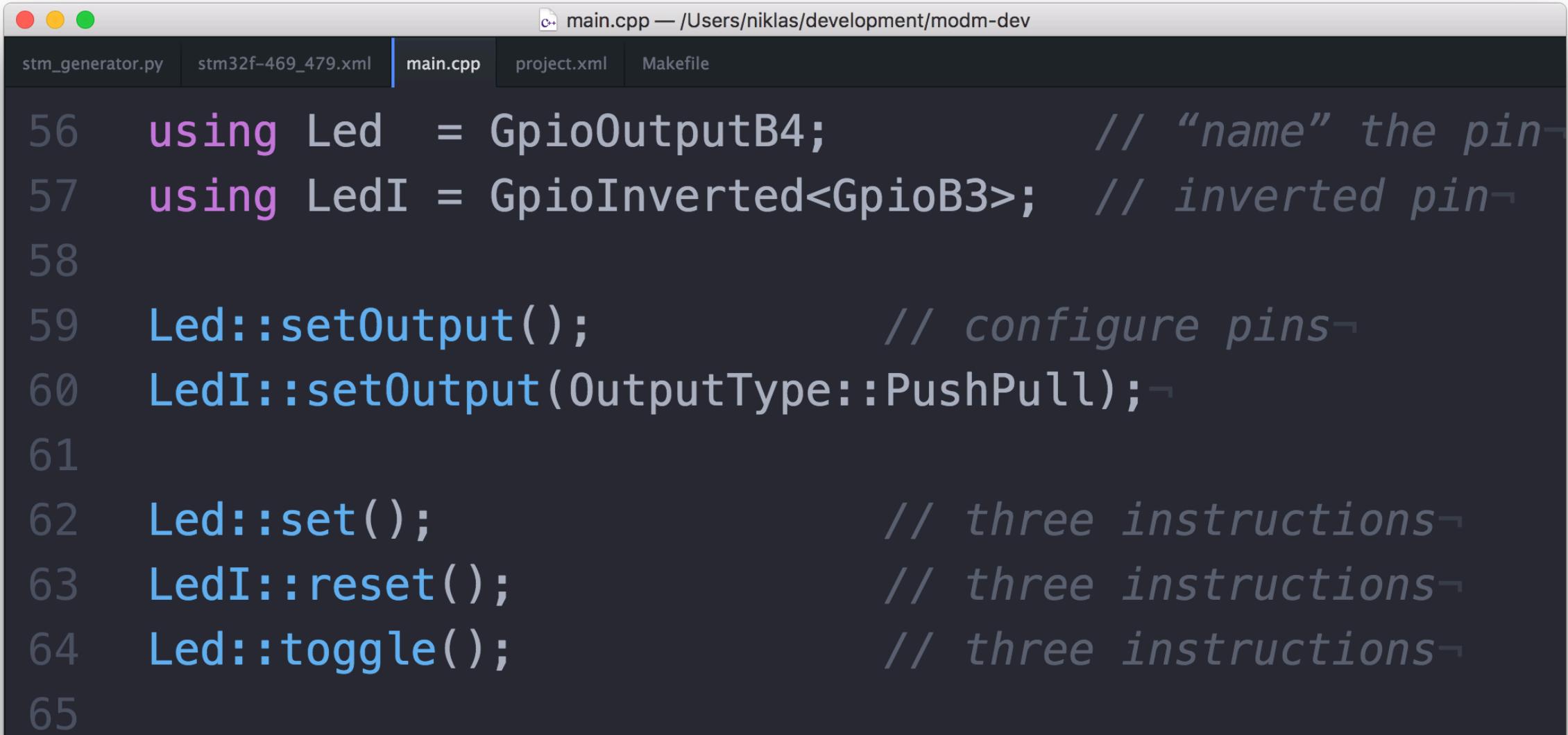
```
? AFIO::REMAP>
[INFO] dfg.memory_query: 2 matches found for AFIO -> .* -> REMAP:
  stm32f[100|101|102|103|105|107][c|r|t|v|z][4|6|8|b|c|d|e|f|g][h|i|t|u|y]
AFIO:MAPR          32-bit @ 0x40010004      all
all:
      31           30           29           28           27           26 (0)        25 (0)        24 (0)
      |             |             |             |             |             |             |             |
      -             -             -             -             -             SWJ_CFG2       SWJ_CFG1       SWJ_CFG0
      |             |             |             |             |             |             |             |
      23           22           21           20           19           18           17           16 (0)
      |             |             |             |             |             |             |             |
      -             -             -             -             -             -             -             TIM5CH4_IREMAP
      |             |             |             |             |             |             |             |
      15 (0)       14           13           12 (0)       11 (0)       10 (0)       9 (0)        8 (0)
      |             |             |             |             |             |             |             |
      PD01_REMAP   -             -             TIM4_REMAP   TIM3_REMAP1  TIM3_REMAP0  TIM2_REMAP1  TIM2_REMAP0
      |             |             |             |             |             |             |             |
      7 (0)        6 (0)        5 (0)        4 (0)        3 (0)        2 (0)        1 (0)        0 (0)
      |             |             |             |             |             |             |             |
      TIM1_REMAP1  TIM1_REMAP0 USART3_REMAP1 USART3_REMAP0 USART2_REMAP  USART1_REMAP  I2C1_REMAP  SPI1_REMAP
      |             |             |             |             |             |             |             |
!f[100|105|107]:
      31           30           29           28           27           26 (0)        25 (0)        24 (0)
      |             |             |             |             |             |             |             |
      -             -             -             -             -             SWJ_CFG2       SWJ_CFG1       SWJ_CFG0
      |             |             |             |             |             |             |             |
      23           22           21           20 (0)      19 (0)      18 (0)      17 (0)      16 (0)
```

# Horizontal porting

	f0	f1	f3	f4	f7
core	✓	✓	✓	✓	✓
gpio	✓	✓	✓	✓	✓
clock	✓	✓	✓	✓	✓

The diagram illustrates the process of horizontal porting across five functional units (f0, f1, f3, f4, f7). Three components are being ported: 'core', 'gpio', and 'clock'. Each component has a row of checkboxes corresponding to the functional units. A blue arrow points to the right at the end of each row, indicating the direction of porting. In the 'core' row, all checkboxes are checked. In the 'gpio' row, all checkboxes are checked except for f0. In the 'clock' row, all checkboxes are checked except for f0.

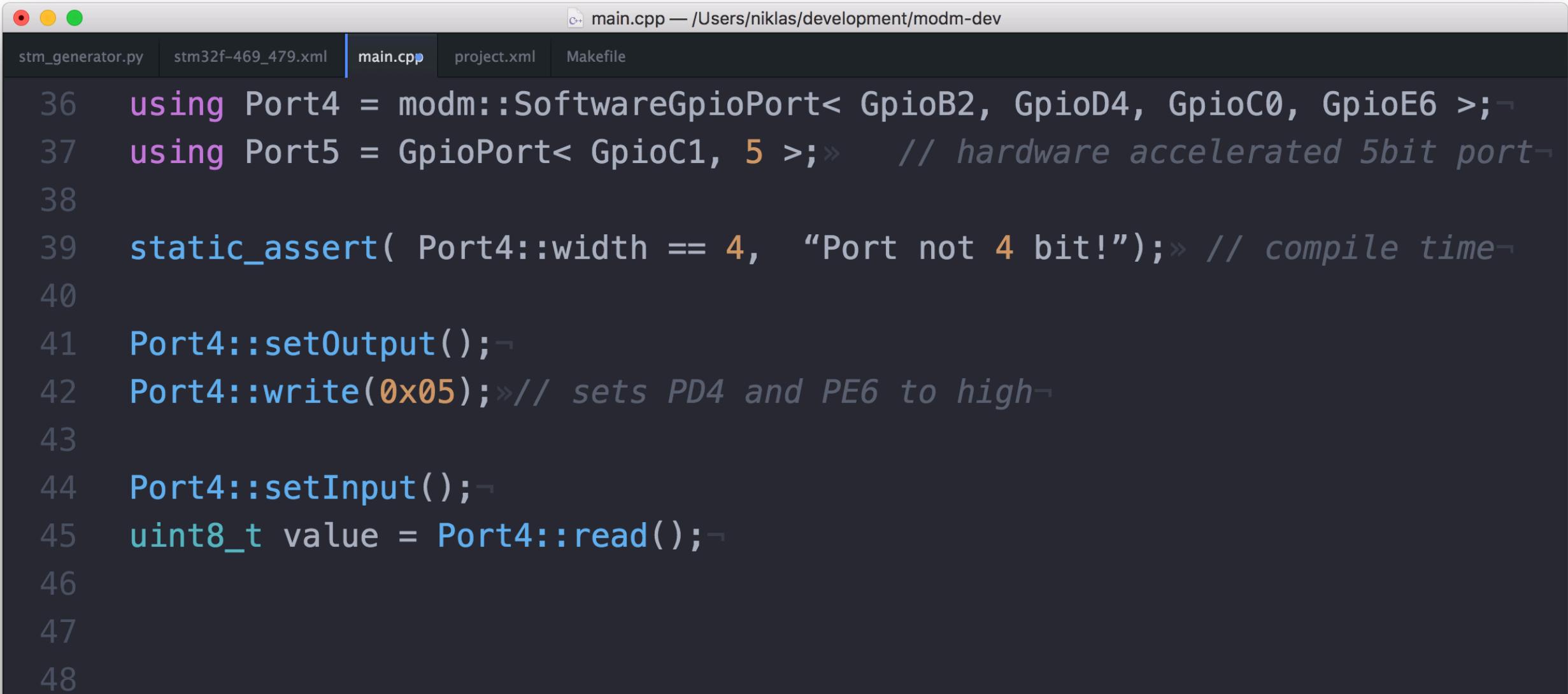
# modm:platform:gpio for STM32



The screenshot shows a macOS terminal window with a dark theme. The title bar reads "main.cpp — /Users/niklas/development/modm-dev". The tab bar includes "stm\_generator.py", "stm32f-469\_479.xml", "main.cpp" (which is selected), "project.xml", and "Makefile". The main pane displays the following C++ code:

```
56     using Led    = GpioOutputB4;           // "name" the pin-
57     using LedI   = GpioInverted<GpioB3>; // inverted pin-
58
59     Led::setOutput();                  // configure pins-
60     LedI::setOutput(OutputType::PushPull);-
61
62     Led::set();                      // three instructions-
63     LedI::reset();                  // three instructions-
64     Led::toggle();                  // three instructions-
65
```

# Compile-time friendly

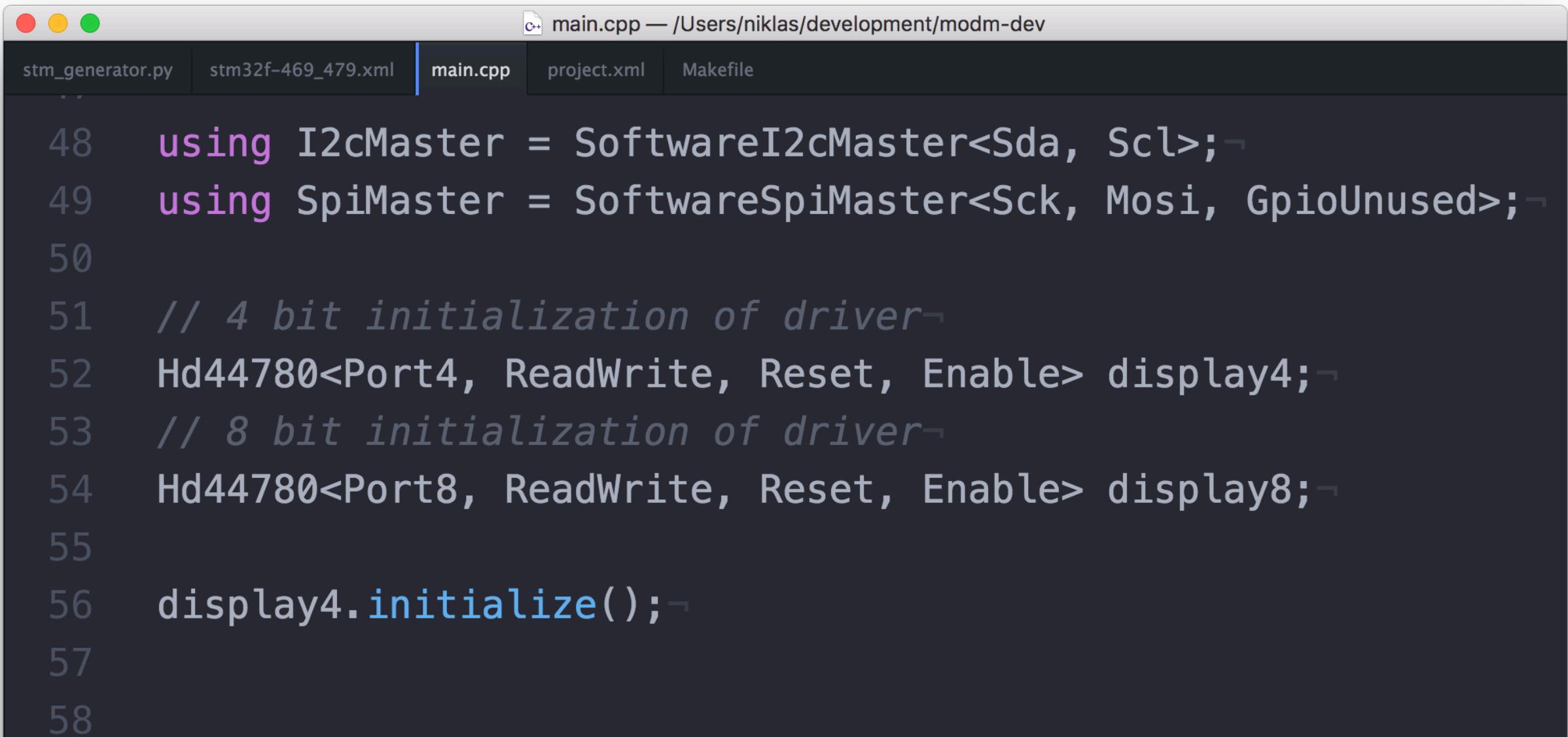


The screenshot shows a code editor window with the following details:

- Title Bar:** main.cpp — /Users/niklas/development/modm-dev
- Tab Bar:** stm\_generator.py, stm32f-469\_479.xml, **main.cpp** (selected), project.xml, Makefile
- Code Area:** Displays the following C++ code with line numbers 36 through 48:

```
36  using Port4 = modm::SoftwareGpioPort< GpioB2, GpioD4, GpioC0, GpioE6 >;  
37  using Port5 = GpioPort< GpioC1, 5 >; // hardware accelerated 5bit port  
38  
39  static_assert( Port4::width == 4, "Port not 4 bit!"); // compile time  
40  
41  Port4::setOutput();  
42  Port4::write(0x05); // sets PD4 and PE6 to high  
43  
44  Port4::setInput();  
45  uint8_t value = Port4::read();  
46  
47  
48
```

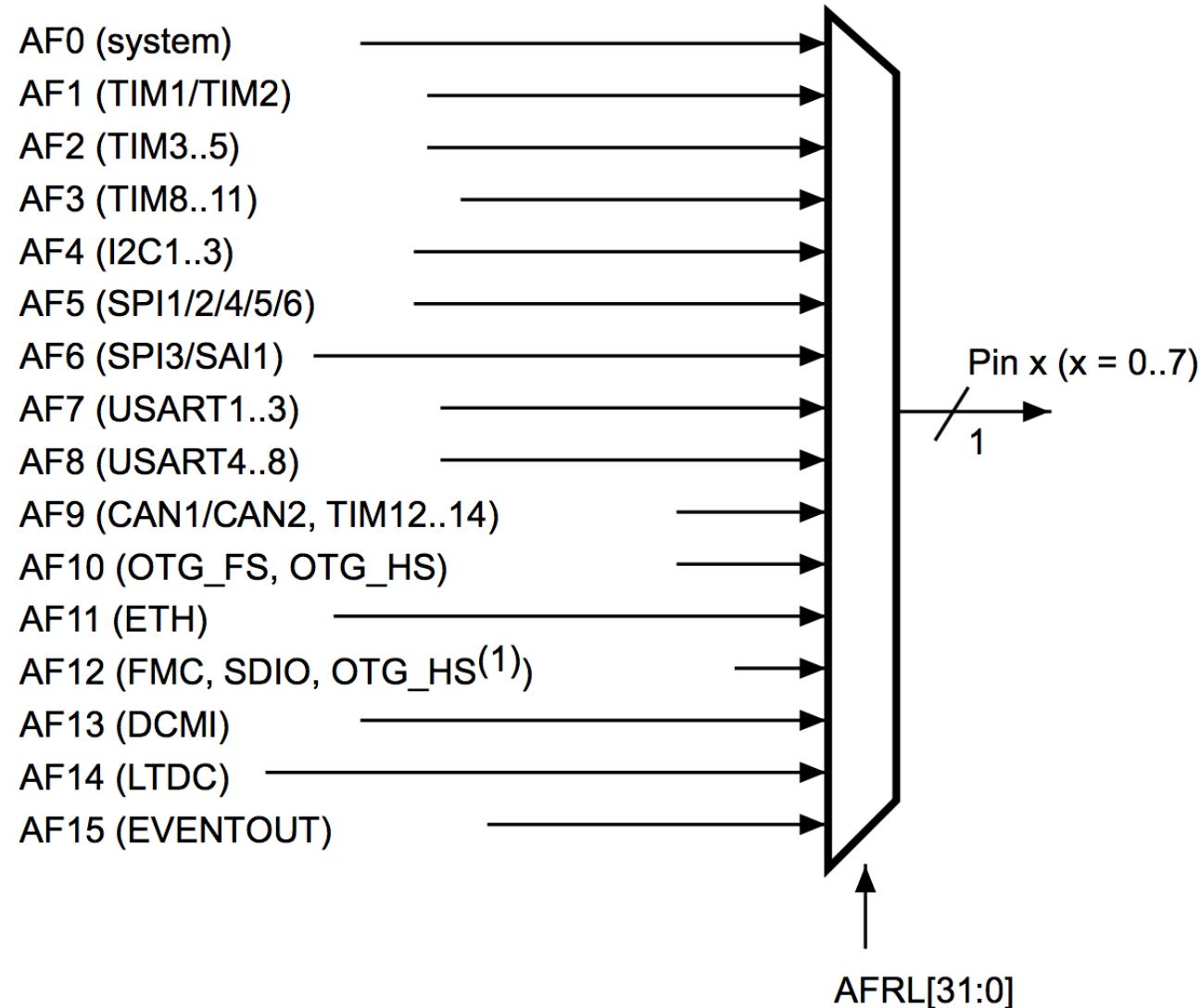
# Bit-bang friendly



A screenshot of a Mac OS X terminal window titled "main.cpp — /Users/niklas/development/modm-dev". The window shows a tab bar with "stm\_generator.py", "stm32f-469\_479.xml", "main.cpp", "project.xml", and "Makefile". The "main.cpp" tab is selected. The terminal content displays C++ code for initializing I2C and SPI drivers using the Modm library. The code includes using statements for SoftwareI2cMaster and SoftwareSpiMaster, and definitions for Hd44780 drivers initialized in 4-bit and 8-bit modes. It concludes with a call to the initialize() method of the display4 object.

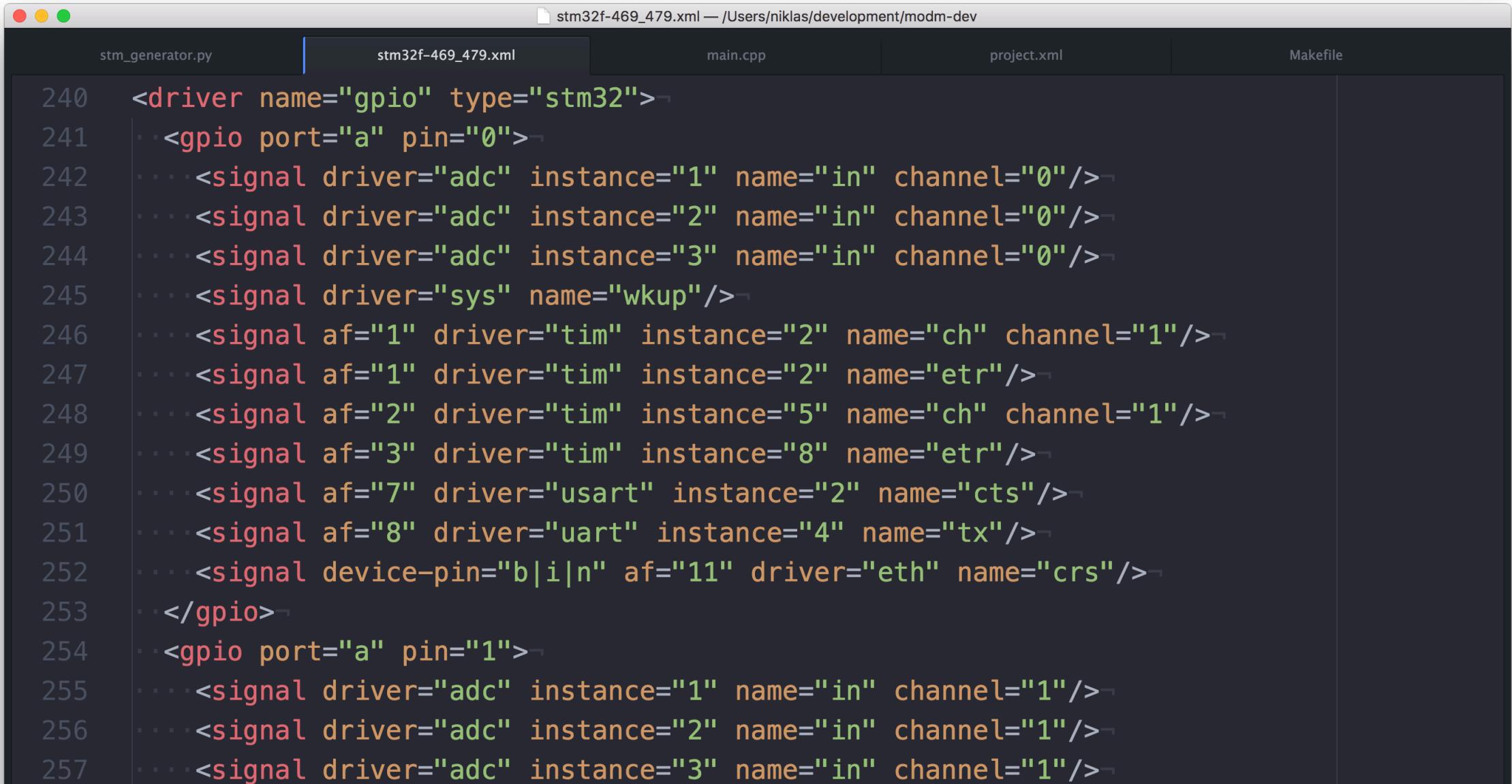
```
48     using I2cMaster = SoftwareI2cMaster<Sda, Scl>;  
49     using SpiMaster = SoftwareSpiMaster<Sck, Mosi, GpioUnused>;  
50  
51     // 4 bit initialization of driver  
52     Hd44780<Port4, ReadWrite, Reset, Enable> display4;  
53     // 8 bit initialization of driver  
54     Hd44780<Port8, ReadWrite, Reset, Enable> display8;  
55  
56     display4.initialize();  
57  
58
```

# STM32 GPIO Alternate Functions





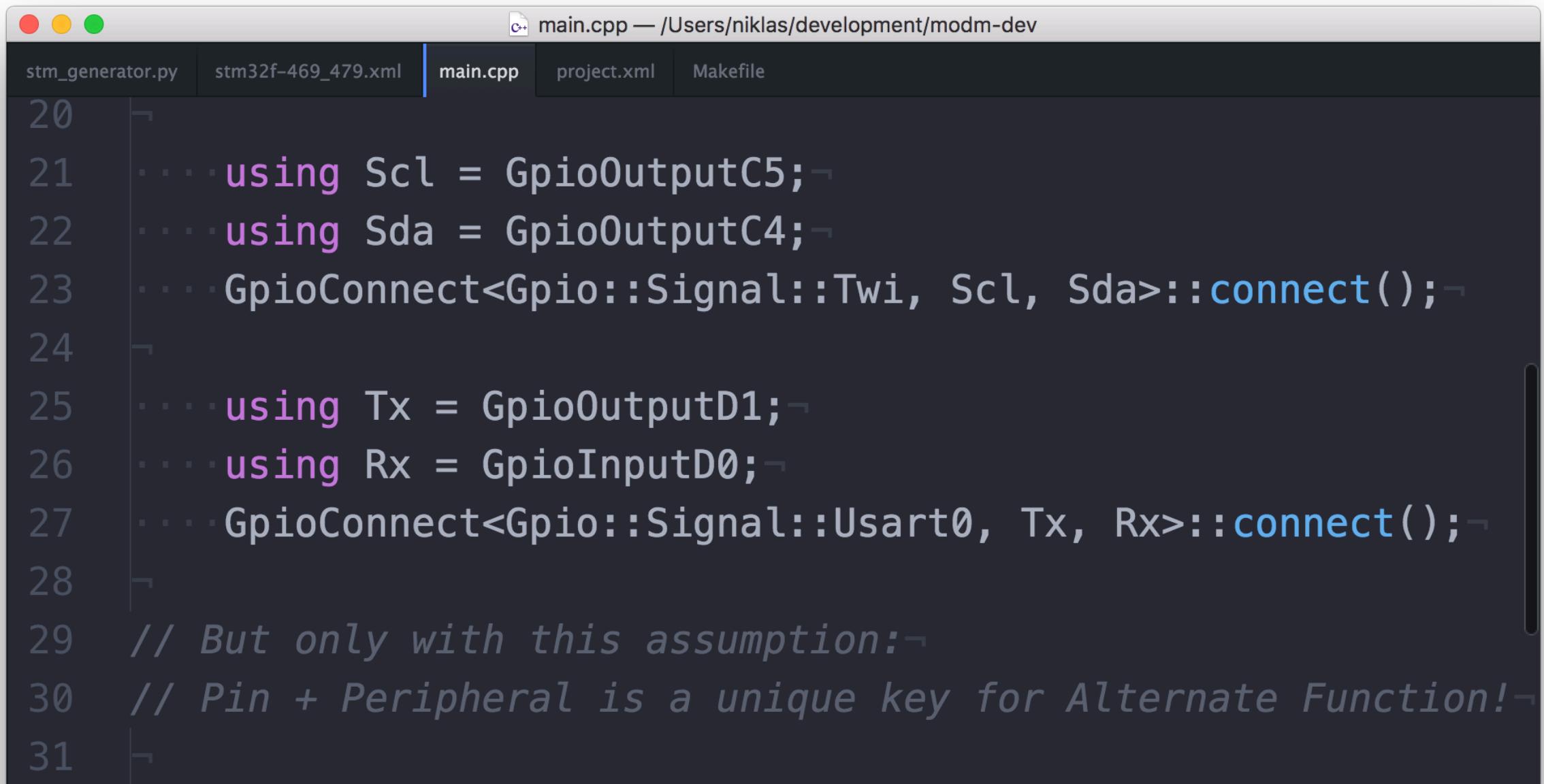
# GPIO data in Device File



The screenshot shows a code editor window with a dark theme. The title bar reads "stm32f-469\_479.xml — /Users/niklas/development/modm-dev". The tabs at the top are "stm\_generator.py", "stm32f-469\_479.xml" (which is active), "main.cpp", "project.xml", and "Makefile". The main editor area displays the following XML code:

```
240 <driver name="gpio" type="stm32">
241   <gpio port="a" pin="0">
242     <signal driver="adc" instance="1" name="in" channel="0"/>
243     <signal driver="adc" instance="2" name="in" channel="0"/>
244     <signal driver="adc" instance="3" name="in" channel="0"/>
245     <signal driver="sys" name="wkup"/>
246     <signal af="1" driver="tim" instance="2" name="ch" channel="1"/>
247     <signal af="1" driver="tim" instance="2" name="etr"/>
248     <signal af="2" driver="tim" instance="5" name="ch" channel="1"/>
249     <signal af="3" driver="tim" instance="8" name="etr"/>
250     <signal af="7" driver="uart" instance="2" name="cts"/>
251     <signal af="8" driver="uart" instance="4" name="tx"/>
252     <signal device-pin="b|i|n" af="11" driver="eth" name="crs"/>
253   </gpio>
254   <gpio port="a" pin="1">
255     <signal driver="adc" instance="1" name="in" channel="1"/>
256     <signal driver="adc" instance="2" name="in" channel="1"/>
257     <signal driver="adc" instance="3" name="in" channel="1"/>
```

# STM32 AF “connector”

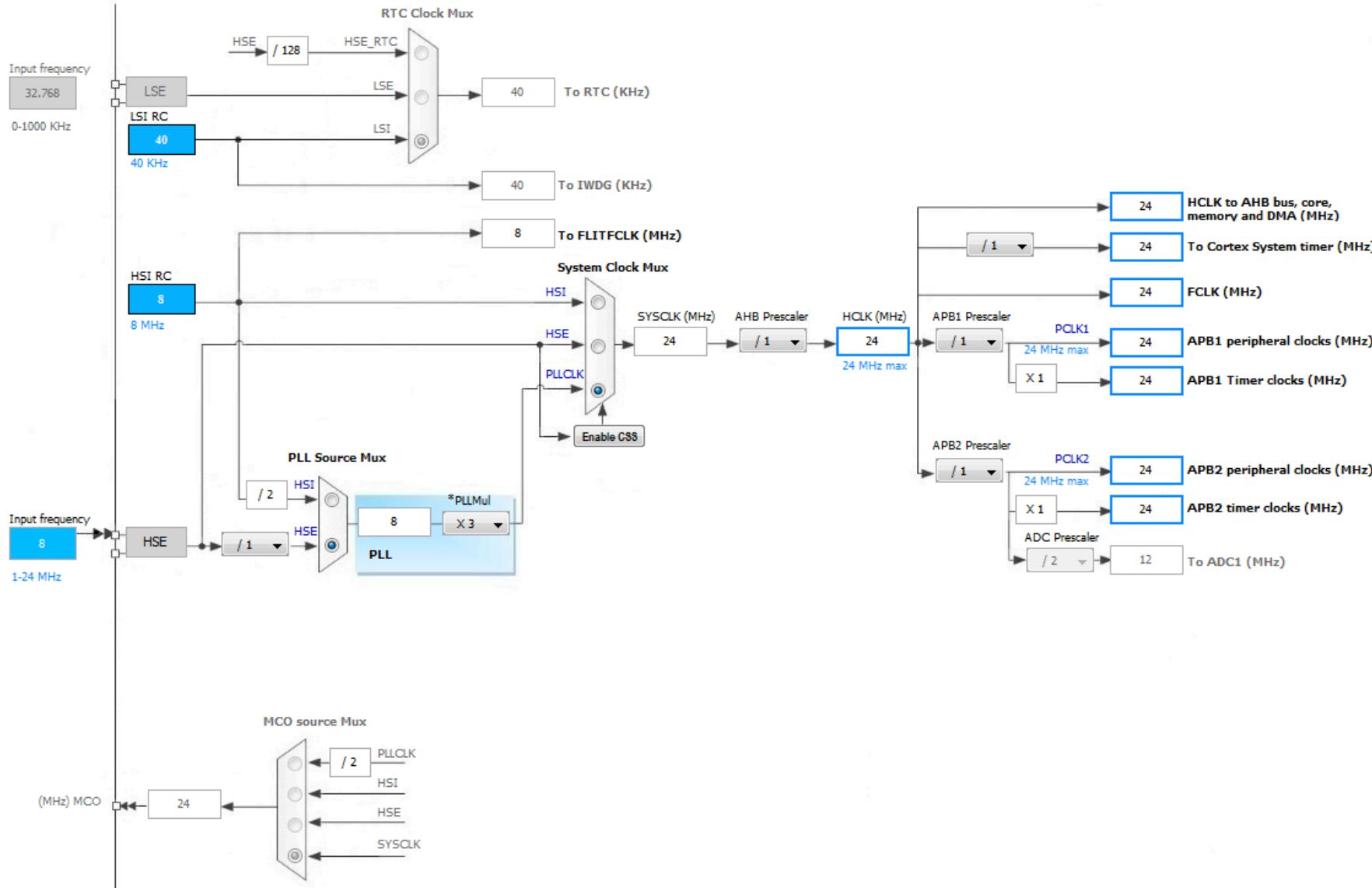


A screenshot of a macOS terminal window titled "main.cpp — /Users/niklas/development/modm-dev". The window shows a code editor with the following C++ code:

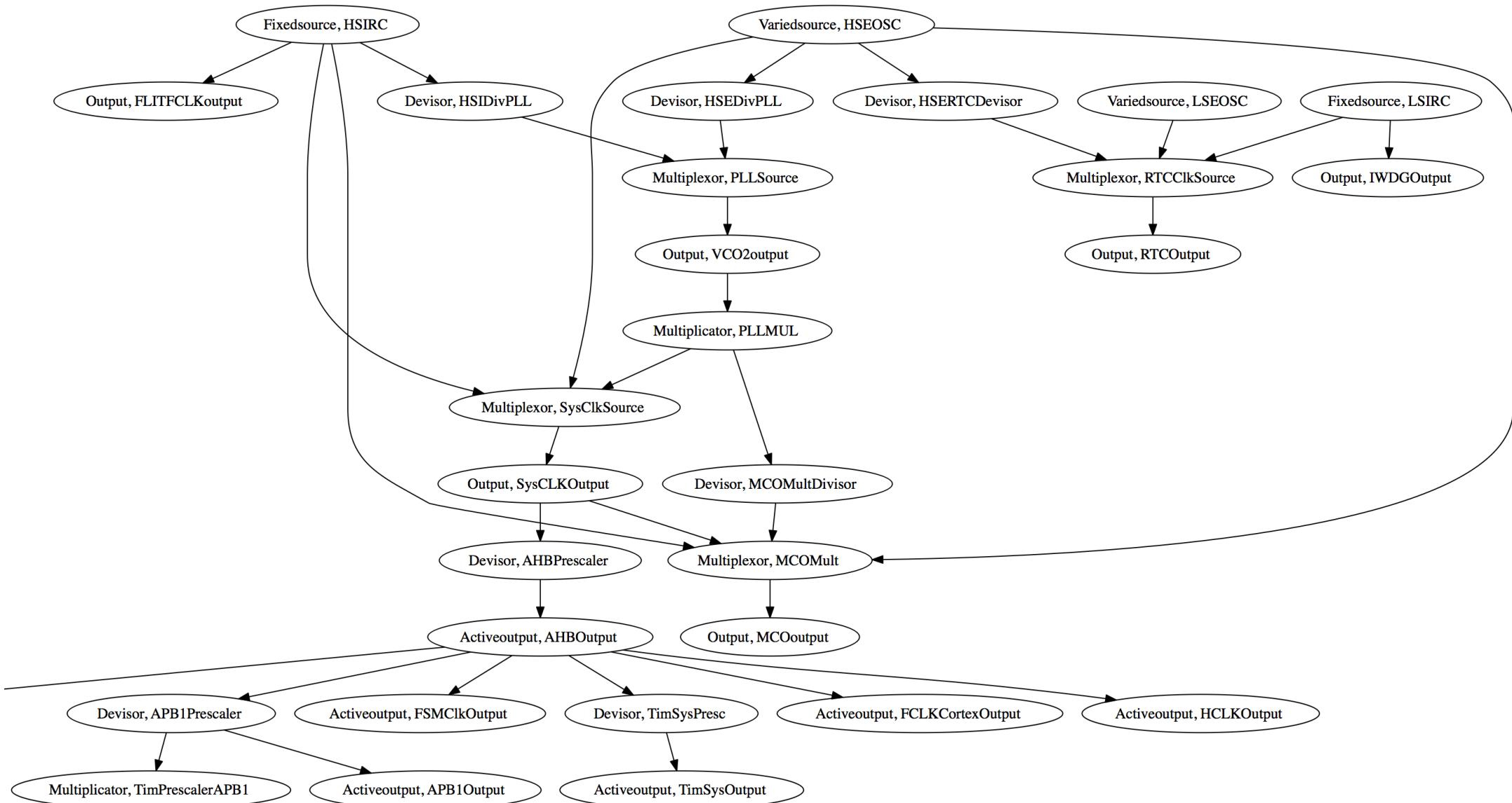
```
20
21     ... using Scl = GpioOutputC5;-
22     ... using Sda = GpioOutputC4;-
23     ... GpioConnect<Gpio::Signal::Twi, Scl, Sda>::connect();-
24
25     ... using Tx = GpioOutputD1;-
26     ... using Rx = GpioInputD0;-
27     ... GpioConnect<Gpio::Signal::Usart0, Tx, Rx>::connect();-
28
29 // But only with this assumption:-
30 // Pin + Peripheral is a unique key for Alternate Function!-
31
```

The code demonstrates the configuration of pins Scl, Sda, Tx, and Rx for I2C and USART0 respectively, using the Modm library's GpioConnect feature. It includes a note about the uniqueness of the pin + peripheral key for alternate functions.

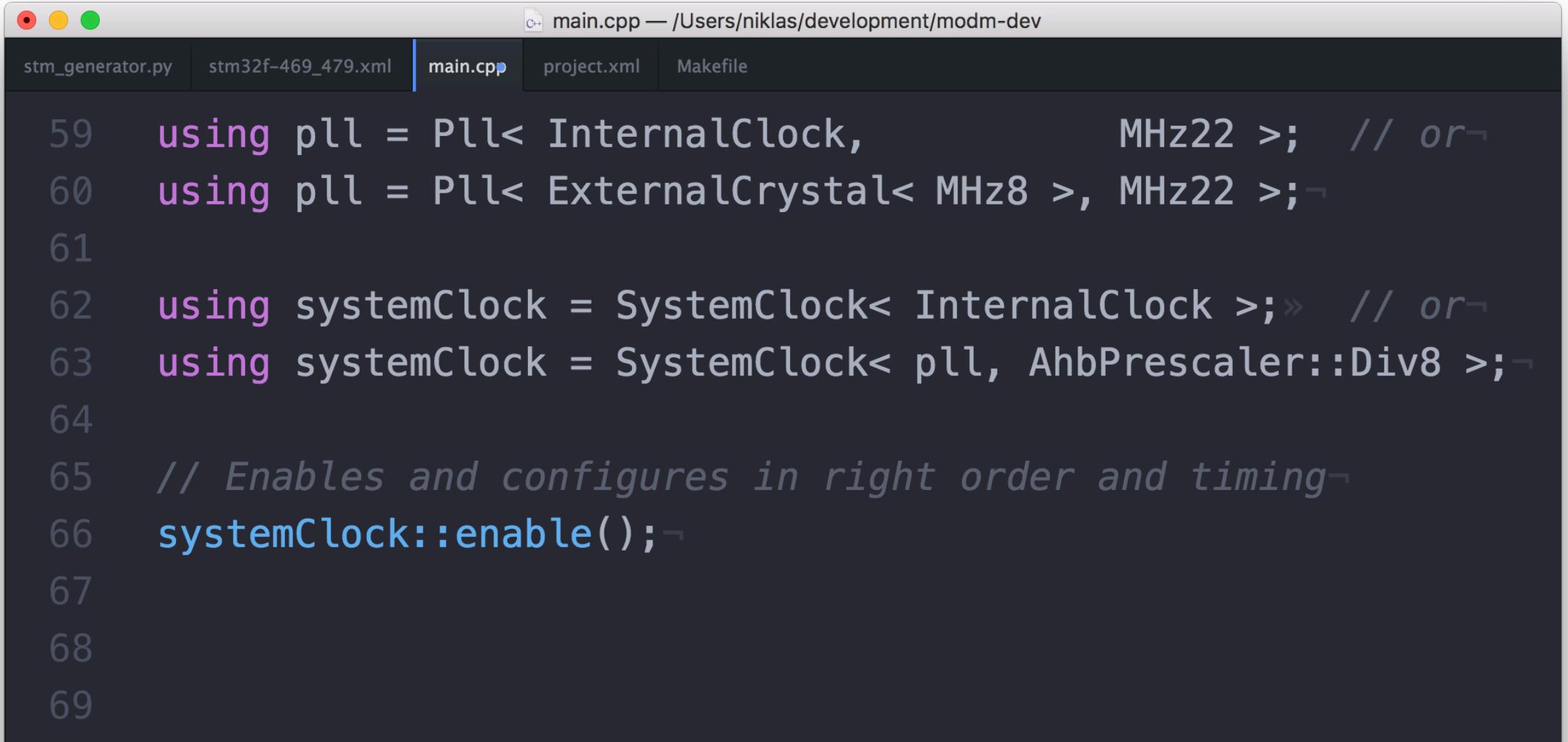
# STM32F100 Clock Tree



# STM32F100 Clock Graph



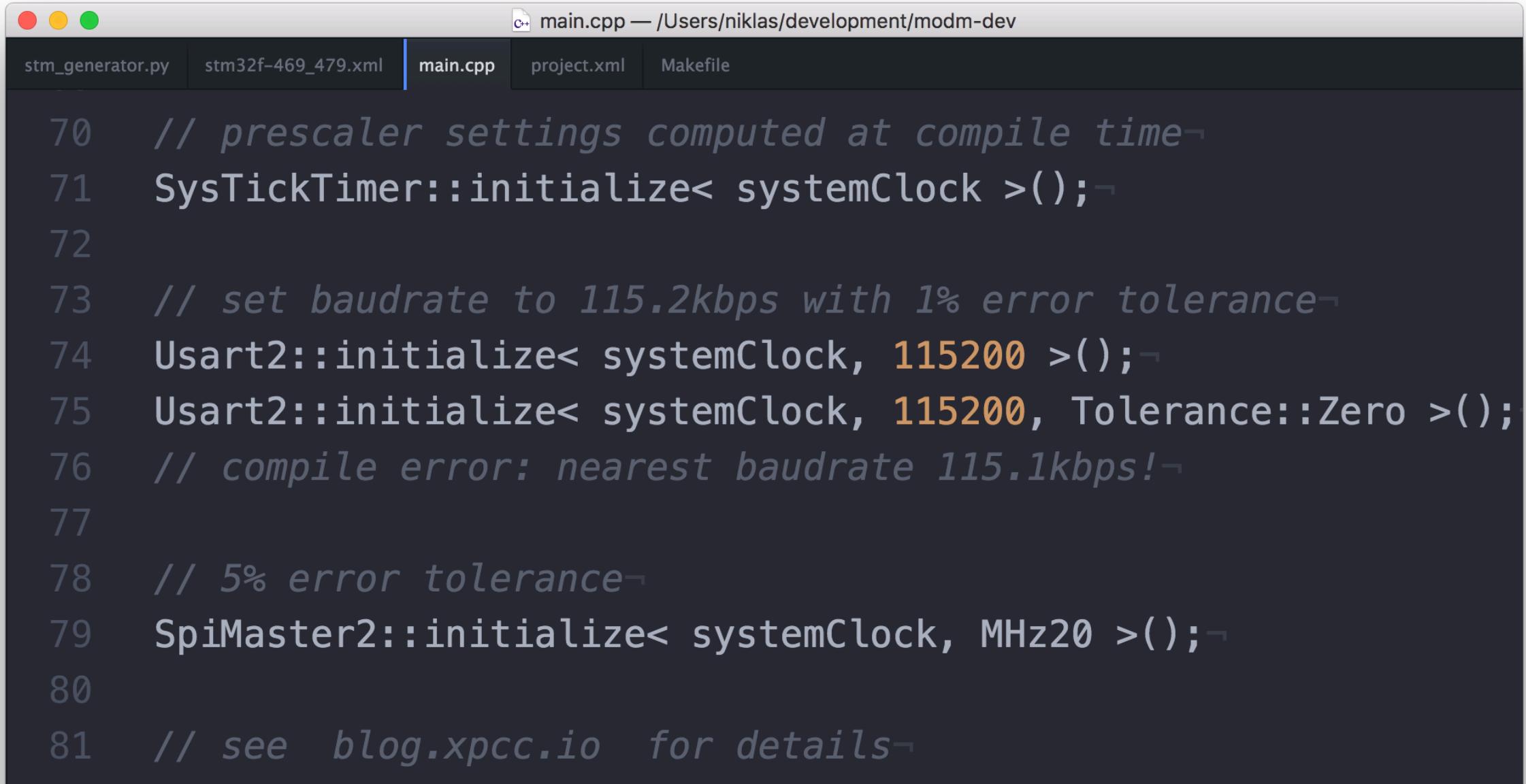
# STM32F100 SystemClock



A screenshot of a macOS terminal window titled "main.cpp — /Users/niklas/development/modm-dev". The window shows a tab bar with "stm\_generator.py", "stm32f-469\_479.xml", "main.cpp", "project.xml", and "Makefile". The "main.cpp" tab is selected. The terminal displays the following C++ code:

```
59     using pll = Pll< InternalClock, MHz22 >; // or-
60     using pll = Pll< ExternalCrystal< MHz8 >, MHz22 >;-
61
62     using systemClock = SystemClock< InternalClock >; // or-
63     using systemClock = SystemClock< pll, AhbPrescaler::Div8 >;-
64
65     // Enables and configures in right order and timing-
66     systemClock::enable();-
67
68
69
```

# Compile-time baudrate computation



The screenshot shows a Mac OS X application window with a dark theme. The title bar reads "main.cpp — /Users/niklas/development/modm-dev". Below the title bar is a navigation bar with five tabs: "stm\_generator.py", "stm32f-469\_479.xml", "main.cpp" (which is selected and highlighted in blue), "project.xml", and "Makefile". The main content area of the window displays the following C++ code:

```
70 // prescaler settings computed at compile time-
71 SysTickTimer::initialize< systemClock >();-
72
73 // set baudrate to 115.2kbps with 1% error tolerance-
74 Usart2::initialize< systemClock, 115200 >();-
75 Usart2::initialize< systemClock, 115200, Tolerance::Zero >();-
76 // compile error: nearest baudrate 115.1kbps!-
77
78 // 5% error tolerance-
79 SpiMaster2::initialize< systemClock, MHz20 >();-
80
81 // see blog.xpcc.io for details-
```

# Thank you for listening!

- Niklas Hauser: [@salkinium](#)
- Fabian Greif: [@dergraaf](#)
- Kevin Läufer: [@ekiwi](#)
- Sascha Schade: [@strongly-typed](#)
- xpcc: [xpcc.io](#)
- modm: [github.com/modm-io](#)